PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-195406

(43)Date of publication of application : 19.07.2001

(51)Int.Cl. G06F 17/30

G06F 12/00

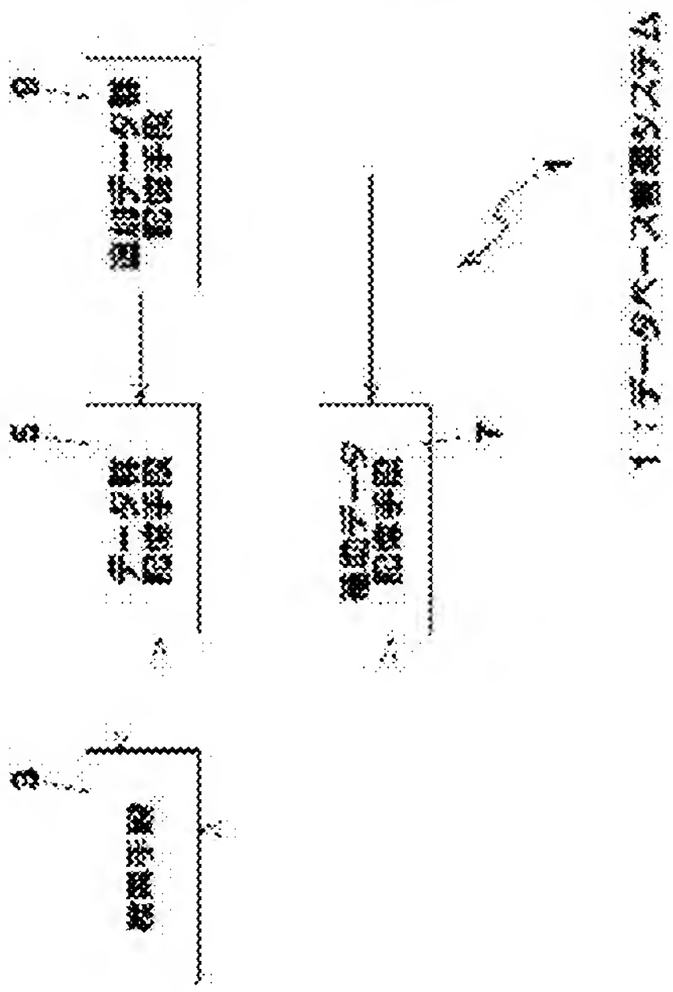
(21)Application number : 2000-000867

(22)Date of filing : 06.01.2000

(71)Applicant : MEDIA FUSION CO LTD
ODA MIKIO

(72)Inventor : SAKAKIBARA ATSUSHI
ODA MIKIO

(54) DATABASE MANAGEMENT SYSTEM



(57)Abstract:

PROBLEM TO BE SOLVED: To provide a database management system, capable of storing XML block data as they are.

SOLUTION: A data group storing means 5 stores data about a plurality of attribute items and also stores a plurality of pieces of block data, in which each attribute item is determined structurally. An auxiliary data storing means 7 stores retrieval auxiliary data including the existence position information of data belonging to each attribute item of the plurality of block data. When a retrieval instruction which includes an attribute item to be retrieved is given, a retrieving means 3 refers to the retrieval auxiliary data stored in the means 7 and retrieves the data stored in the means 5. Thus, it is possible to perform high-speed data retrieval even

though the plurality of block data, in which each attribute item is structurally determined are stored as they are.

LEGAL STATUS

[Date of request for examination] 22.09.2000

[Date of sending the examiner's decision of rejection] 11.08.2003

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *
JPO and NCIPI are not responsible for any damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The XML block data with which data are memorized about two or more attribute items, and each attribute item consisted of tree structures While it is the

database management system managed using a computer, and memorizing said XML block data for every XML block data to the 1st storage region, with the structure concerned held If the index data for retrieval which memorize data's belonging to each attribute item of each XML block data existence positional information collectively for every attribute are memorized to the 2nd storage region and new XML block data is given While carrying out the additional storage of performing the following processings and the A aforementioned XML block data in said 1st storage region, with the tree structure concerned held B) about the attribute item which is in agreement with the attribute item of said index data for retrieval among the attribute items of said newly given XML block data About the attribute item which adds data's of attribute item concerned existence positional information to said 2nd storage region, and does not exist in the attribute item of said index data for retrieval The database management system characterized by ** which adds data's of attribute item concerned existence positional information to said 2nd storage region while adding to said 2nd storage region as a new attribute item.

[Claim 2] Are the data constellation data were remembered to be about two or more attribute items, and the data constellation as which each attribute item is determined structurally A data constellation storage means to memorize [two or more], without breaking down the structure concerned, an auxiliary-data storage means to memorize the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute, If the data constellation data were remembered to be about two or more attribute items is newly given, while carrying out additional storage, for said data storage means About the attribute item which is in agreement with the attribute item of said retrieval auxiliary data among the attribute items of said newly given data constellation About the attribute item which adds data's of attribute item concerned existence positional information, and does not exist in the attribute item of said retrieval auxiliary data The database management system equipped with an additional data group processing means to add data's of attribute item concerned existence positional information while adding as a new attribute item.

[Claim 3] In the database management system of claim 2, it is characterized by the structure of each of said attribute item being a tree structure.

[Claim 4] Are the data constellation data were remembered to be about two or more attribute items, and the data constellation as which each attribute item is determined structurally A data constellation storage means to memorize [two or more], without breaking down the structure concerned, an auxiliary-data storage means to memorize

the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute, The database management system equipped with a retrieval means to search the data which made reference the retrieval auxiliary data memorized by said auxiliary-data storage means, and were memorized by the data storage means when the retrieval instruction including the attribute item to search was given.

[Claim 5] In the database management system of claim 4, said auxiliary-data storage means has memorized the structure of each of said attribute item, and said retrieval means is characterized by searching the data which made structure of each attribute item reference and were memorized by the data storage means.

[Claim 6] A) the data constellation storage region and B which are the data constellation data were remembered to be about two or more attribute items, and memorize two or more data constellations as which each attribute item is determined structurally, without breaking down the structure concerned -- the data-logging medium which has the auxiliary-data storage region which memorizes collectively the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information for every attribute.

[Claim 7] It is the data retrieval approach of searching data using a computer, and is the data constellation data were remembered to be about two or more attribute items. The data constellation as which each attribute item is determined structurally, without breaking down the structure concerned Plurality, The retrieval auxiliary data which memorizes to the 1st storage region and memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute is memorized to the 2nd storage region. The data retrieval approach which will make said retrieval auxiliary data reference and will be characterized by searching the data constellation memorized in said 1st storage region if the retrieval instruction including the attribute item to search is given.

[Claim 8] It is the data control approach using a computer, and is the data constellation data were remembered to be about two or more attribute items. Two or more data constellations as which each attribute item is determined structurally are memorized to the 1st storage region, without breaking down the structure concerned. If the data constellation the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute is remembered to be to the 2nd storage region, and data were remembered to be about two or more attribute items is newly

given While carrying out additional storage in said 1st storage region, about the attribute item which is in agreement with the attribute item of said retrieval auxiliary data among the attribute items of said newly given data constellation About the attribute item which adds data's of attribute item concerned existence positional information, and does not exist in the attribute item of said retrieval auxiliary data The data control approach characterized by adding data's of attribute item concerned existence positional information to said 2nd storage region while adding as a ******** attribute item.

[Claim 9] While memorizing two or more data constellations as which it is the data constellation data were remembered to be about two or more attribute items, and each attribute item is beforehand determined to the 1st storage region structurally, without breaking down the structure concerned It is the storage which memorized the program which realizes the computer which memorized the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute to the 2nd storage region as database management equipment. The storage which memorized the program for making said computer perform processing which searches the data constellation which made said retrieval auxiliary data reference and was memorized in said 1st storage region when the retrieval instruction including the attribute item to search is given.

[Translation done.]

* NOTICES *

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] Especially this invention relates to improvement in the degree of freedom of DS about a database management system.

[0002]

[Description of the Prior Art] Generally, in the database management system, it has two or more records and each record consists of two or more fields. The DS (data type, data length, etc.) of this field is defined beforehand. Thus, by defining the data of each field beforehand, high-speed search becomes possible.

[0003] By the way, XML data attract attention as structural description mold data today. Since these XML data can make an attribute add to the data itself, the given data can carry out automatic extracting of whether it is data to which field. For example, if it is XML data shown in drawing 3 , data "Ichiro Yamada" can be memorized in the field "an identifier", and a computer can make the field "an executive" memorize data "a section chief" by referring to the tag of each item.

[0004]

[Problem(s) to be Solved by the Invention] However, there were the following problems in the above-mentioned database management system. When each record is made into a fixed length, the number of the fields and field length for every record turn into a fixed length. Therefore, DS must be reconstructed when newly changing an addition and field length of the field. That is, even if XML data including a new attribute are given, the data of the attribute concerned are disregarded. The amount of data which a database management system manages has many huge things, for example, it becomes very complicated processing to change about all DS for the new field which only one record of the 1 million records has.

[0005] Moreover, even when data do not exist in the field concerned, the data input of the data (null etc.) set by the DS concerned must be carried out, and a useless field occurs.

[0006] Although making each record into variable length is also considered in order to solve this problem, it is necessary to define the number of the fields of each record beforehand even in this case only by the ability to make field length of each field into variable length. Therefore, the addition of the new field cannot be performed.

[0007] That is, in the conventional database management system, it had not become the structure of utilizing enough the merit of the XML data that an attribute can be added to the data itself. Moreover, although XML data could describe the relational structure of each item using said tag, in the conventional database management system, this relational structure was not memorizable as it was.

[0008] This invention solves the above-mentioned problem and it aims at offering the

flexible database management system which can fully employ that property efficiently about the description mold data of an attribute good addition mold.

[0009]

[Means for Solving the Problem] 1) The database management system concerning this invention The XML block data with which data are memorized about two or more attribute items, and each attribute item consisted of tree structures While it is the database management system managed using a computer, and memorizing 1 aforementioned XML block data for every XML block data to the 1st storage region, with the structure concerned held the index data for retrieval which memorize data's belonging to each attribute item of each XML block data existence positional information collectively for every attribute -- the 2nd storage region -- memorizing -- 2, if new XML block data is given While carrying out the additional storage of performing the following processings and the A aforementioned XML block data in said 1st storage region, with the tree structure concerned held B) about the attribute item which is in agreement with the attribute item of said index data for retrieval among the attribute items of said newly given XML block data About the attribute item which adds data's of attribute item concerned existence positional information to said 2nd storage region, and does not exist in the attribute item of said index data for retrieval While adding to said 2nd storage region as a new attribute item, data's of attribute item concerned existence positional information is added to said 2nd storage region. Therefore, the XML block data with which each attribute item consisted of tree structures can be memorized, without breaking down the structure concerned. Furthermore, additional XML block data can be memorized altogether, without breaking down the structure concerned. Moreover, since said index data for retrieval are gathered for every attribute, the data of a certain attribute can be searched at a high speed. The database management system which can utilize by this the property of the XML data that an attribute can be added to each data can be offered.

[0010] 2) In the database management system concerning this invention 1) It is the data constellation data were remembered to be about two or more attribute items. A data constellation storage means to memorize two or more data constellations as which each attribute item is determined structurally, without breaking down the structure concerned, 2) An auxiliary-data storage means to memorize the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute, 3) If the data constellation data were remembered to be about two or more attribute items is newly given, while carrying out additional storage, for said data

storage means About the attribute item which is in agreement with the attribute item of said retrieval auxiliary data among the attribute items of said newly given data constellation Data's of attribute item concerned existence positional information was added, and about the attribute item which does not exist in the attribute item of said retrieval auxiliary data, while adding as a new attribute item, it has an additional data group processing means to add data's of attribute item concerned existence positional information.

[0011] Therefore, the data constellation as which each attribute item is determined structurally can be memorized, without breaking down the structure concerned. Furthermore, an additional data constellation can be memorized altogether, without breaking down the structure concerned. Moreover, since said retrieval auxiliary data is gathered for every attribute, the data of a certain attribute can be searched at a high speed. The database management system which can utilize by this the property of the attribute addition mold data that an attribute can be added to each data can be offered.

[0012] 3) In the database management system concerning this invention, the structure of each of said attribute item is a tree structure. Therefore, the data constellation which has the attribute item of a tree structure is manageable as a database.

[0013] 4) The database management system concerning this invention 1) It is the data constellation data were remembered to be about two or more attribute items. A data constellation storage means to memorize two or more data constellations as which each attribute item is determined structurally, without breaking down the structure concerned, 2) If the retrieval instruction including an auxiliary-data storage means to memorize the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute, and the attribute item which carries out 3 retrieval is given The retrieval auxiliary data memorized by said auxiliary-data storage means was made reference, and it has a retrieval means to search the data memorized by the data storage means. Thus, even when the data constellation structurally constituted by searching the desired data with which each attribute item exists in the data constellation defined structurally using said retrieval auxiliary data is memorized as it was, data retrieval becomes possible at a high speed.

[0014] 5) In the database management system concerning this invention, said auxiliary-data storage means has memorized the structure of each of said attribute item, and said retrieval means searches the data which made structure of each

attribute item reference and were memorized by the data storage means. Therefore, even if only the attribute item searched as a retrieval instruction is given, high-speed search becomes possible.

[0015] 6) In the data-logging medium which recorded the data concerning this invention A) It is the data constellation data were remembered to be about two or more attribute items. It has the auxiliary-data storage region which memorizes collectively the retrieval auxiliary data which memorizes data's which belong data constellation's as which each attribute item's is determined structurally to data constellation storage region and each attribute item's of data constellation of B aforementioned plurality which are memorized without breaking down structure's concerned existence positional information for every attribute. Therefore, even if it memorizes two or more data constellations as which each attribute item is determined structurally, without breaking down the structure concerned, they can be searched at high speed using said retrieval auxiliary data.

[0016] 7) In the data retrieval approach of searching data using the computer concerning this invention 1) It is the data constellation data were remembered to be about two or more attribute items. The data constellation as which each attribute item is determined structurally, without breaking down the structure concerned Plurality, The retrieval auxiliary data which memorizes to the 1st storage region and memorizes data's belonging to each attribute item of two or more 2 aforementioned data constellations existence positional information collectively for every attribute is memorized to the 2nd storage region. 3) If the retrieval instruction including the attribute item to search is given, said retrieval auxiliary data will be made reference and the data constellation memorized in said 1st storage region will be searched. Thus, data retrieval becomes possible from the data constellation constituted structurally at a high speed by searching the desired data with which each attribute item exists in the data constellation defined structurally using said retrieval auxiliary data.

[0017] 8) In the data control approach using the computer concerning this invention Are the data constellation data were remembered to be about two or more attribute items, and the data constellation as which each attribute item is determined structurally The retrieval auxiliary data which memorizes more than one to the 1st storage region, without breaking down the structure concerned, and memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute is memorized to the 2nd storage region. If the data constellation data were remembered to be about two or more attribute items is newly given, while carrying out additional storage in said 1st storage

region About the attribute item which is in agreement with the attribute item of said retrieval auxiliary data among the attribute items of said newly given data constellation Data's of attribute item concerned existence positional information is added, and about the attribute item which does not exist in the attribute item of said retrieval auxiliary data, while adding as a ******** attribute item, data's of attribute item concerned existence positional information is added to said 2nd storage region. Therefore, two or more data constellations as which each attribute item is determined structurally can be memorized, without breaking down the structure concerned. Moreover, an additional data constellation can be memorized altogether, without breaking down the structure concerned.

[0018] Although XML data were used for "the description mold data of an attribute good addition mold" with the operation gestalt, as long as it is data which are not limited to this but can add an attribute, they may be what kind of thing. Moreover, you may be except the method specified with a tag like [technique / of attribute addition] XML data.

[0019]

[Embodiment of the Invention] 1. Explain 1 operation gestalt of explanation this invention of a functional block diagram based on a drawing. The database management system 1 shown in drawing 1 is equipped with the data constellation storage means 5, the auxiliary-data storage means 7, the retrieval means 3, and the additional data group processing means 9.

[0020] The data constellation storage means 5 is the data constellation data were remembered to be about two or more attribute items, and they are memorized, without breaking down the structure concerned for the data constellation as which each attribute item is determined structurally. [two or more] The auxiliary-data storage means 7 memorizes the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute. If the retrieval instruction including the attribute item to search is given, the retrieval means 3 will make reference the retrieval auxiliary data memorized by the auxiliary-data storage means 7, and will search the data memorized by the data storage means 5.

[0021] Therefore, data retrieval becomes possible from the data constellation constituted structurally at a high speed. Moreover, since the retrieval auxiliary data was gathered for every attribute and said existence positional information is memorized, high-speed search becomes more possible.

[0022] If the data constellation data were remembered to be about two or more

attribute items is newly given, while the additional data group processing means 9 carries out additional storage, for the data storage means 5 About the attribute item which is in agreement with the attribute item of said retrieval auxiliary data among the attribute items of said newly given data constellation Data's of attribute item concerned existence positional information is added, and about the attribute item which does not exist in the attribute item of said retrieval auxiliary data, while adding as a ******** attribute item, the additional storage of the data's of attribute item concerned existence positional information is carried out at said auxiliary-data storage means 7. Therefore, two or more data constellations as which each attribute item is determined structurally can be memorized as they are, without breaking down the structure concerned. Moreover, an additional data constellation can be memorized altogether, without breaking down the structure concerned.

[0023] 2. An example of a hardware configuration which realized the database management system 1 shown in drawing 1 to hardware configuration drawing 2 using CPU is shown. The database management system 1 is equipped with CPU23, memory 27, a hard disk 26, FDD25, the monitor 32, the keyboard 28, the mouse 31, and the bus line 29.

[0024] CPU23 controls each part through a bus line 29 according to the program memorized by the hard disk 26. This program is read from flexible disk 25a the program was remembered to be through FDD25, and is installed on a hard disk 26. In addition, you may make it make the program of CD-ROM, an IC card, etc. install on a hard disk from the computer-readable record medium unified ontically in addition to a flexible disk. Furthermore, you may make it download using a communication line.

[0025] He is trying to make a computer execute indirectly the program stored in the flexible disk in this operation gestalt by making a program install on a hard disk 26 from a flexible disk. However, it may be made to perform the program stored in the flexible disk directly from FDD25, without being limited to this. In addition, by computer, the thing in which immediate execution is possible only by [as it is] installing as a program which can be executed combines with a thing (for example, that by which the data compression is carried out is thawed) to be changed, and the module part of further others, and of course once contains in other gestalten etc. what can be performed.

[0026] In a hard disk 26, a control program and an operating system (OS) are memorized. As an operating system, WindowsNT (trademark) by Microsoft [Corp.] Corp. is employable, for example.

[0027] Container box 26c and index data 26i for retrieval are further memorized by the

hard disk 26. Two or more storage of the XML data with which the attribute was added to each data as shown in drawing 3 is carried out at container box 26c, not breaking down the attribute-structure relation. When it explains concretely, the XML data shown in drawing 3 A are expressed with a tree structure as shows each attribute-structure relation to drawing 4 . In order to memorize this tree structure as it is, as shown in drawing 5 , the pointer in which it is shown to which field the live data about each attribute are memorized is memorized. For example, about the attribute "an identifier", the pointer P1 in which the start address of live data "Ichiro Yamada" is shown is memorized. Moreover, in this operation gestalt, since each attribute-structure relation is expressed with the tree structure The attribute "the section" belongs to the attribute "affiliation" further, and the live data are memorized by the pointer P5. Furthermore, the attribute "a section" is memorized in the field to which it points with a pointer P5 in this case, and the live data of an attribute "a section" are memorized to the field to which it points with a pointer P6. In this operation gestalt, the data constellation of the XML data of 1 is called block data. Two or more storage of this block data is carried out at container box 26c.

[0028] Index data 26for retrieval i is explained using drawing 7 . Index data 26i for retrieval is index data for searching two or more block data memorized by container box 26c, and is memorized by the integrated tree structure which unified the tree structure of each attribute of two or more block data memorized by container box 26c. Furthermore, they are collectively memorized by Container id and the start address (index information) of live data for every attribute about the block data which has [attributes / of an integrated tree structure / all] data about the attribute concerned.

[0029] An integrated tree structure is explained. An integrated tree structure means the tree structure with which the tree structure of all block data was unified by having performed the OR operation about the tree structure of all block data. For example, as shown in drawing 6 , when there are a tree structure shown in drawing 6 A, a tree structure shown in drawing 6 B, and a tree structure shown in drawing 6 C, drawing 6 D serves as an integrated tree structure.

[0030] Block id and the start address (index information) of the live data of block data which have data for the attribute concerned are memorized by each attribute of this integrated tree structure. for example, the address 0001 in the block id with which the live data "Ichiro Yamada" of an attribute [in / in an attribute "an identifier" / in an integrated tree structure / the block data of block id=0] "an identifier" are stored, and a container box and the address 0021 in block id=1 in which the live data "Jiro Tanaka" of the attribute "an identifier" in the block data of block id=1 are stored, and

a container box .... as -- the data for retrieval are memorized. About how to use this address for retrieval, it mentions later.

[0031] In addition, since it is an integrated tree structure, it may not have [block data / a certain] data about the attribute concerned. For example, since the data of block id=2 do not exist in an attribute "a section" when an attribute "a section" does not exist [the data of block id=2] in an attribute "affiliation" with an attribute "the Administrative Division", the index is not memorized.

[0032] Thus, the structure of each attribute item is a tree structure, and as for the block data memorized by container box 26c, they are collectively memorized by index data for data to search the existing block data for every attribute about each attribute.

[0033] Various kinds of results of an operation etc. are memorized by memory 27.

[0034] A keyboard 28 and a mouse 31 are input means to input various kinds of data or an instruction. A retrieval result etc. is displayed on a monitor 32.

[0035] 3. Explain flow chart (3.1) data addition storage processing using drawing 8 about the data storage processing to a hard disk 26. While block data with a data addition new to container box 26c is added, index data are added to index 26i for retrieval. Here, when the integrated tree structure shown in drawing 4 was already memorized, the XML data shown in drawing 9 A should be given. CPU23 memorizes the given XML data to container box 26c, not breaking down the attribute structure (step S1). Thereby, one block data is added.

[0036] CPU23 extracts the tree structure shown in drawing 9 B from these XML data (step S3). This is realizable by extracting the relational structure of the tag of XML data. For example, it turns out that a tag "an identifier", "affiliation", an "executive", and the "telephone number" are located in the low order of a tag "personnel" from drawing 9 A, the tag "the section" is further located in the low order of a tag "an executive", and the tag "a section" is further located in the low order of a tag "the section." Thereby, as shown in drawing 9 B, the tree structure of each attribute can be extracted.

[0037] CPU23 initializes the processing number i (step S5), and makes the i-th attribute an attention attribute (step S7). It judges whether CPU23 already exists in an integrated tree structure about an attention attribute (step S9). In this case, since an attribute "an identifier" already exists in the integrated tree structure of drawing 7 , it progresses to step S13 and the start address in said container box of the data "Taro Suzuki" belonging to the attribute concerned is memorized with the block id concerned for the attribute "an identifier" of said integrated tree structure.

[0038] CPU23 judges whether the attention attribute which is not examined exists

(step S15), when it exists, increments the processing number i (step S17), and performs processing not more than step S7. In this case, since it already exists altogether about the remaining attribute "affiliation", the "section", a "section", an "executive", and the "telephone number", only new address data are memorized similarly (step S13), and processing will be ended if all items are examined.

[0039] In addition, what is necessary is just to judge that whether an attribute exists or not does not already exist when the attributes of the high order differ. Namely, what is necessary is just to judge that it already exists in full coincidence. For example, what is necessary is just to judge that it is a different attribute, when an attribute "a section" exists under affiliation although an attribute "a section" is located in the lower part of an attribute "the section" in this case.

[0040] Thus, without breaking down the attribute-structure relation, new XML data are memorized and additional storage is further carried out at index 26i for retrieval.

[0041] The case where the attribute which is not an attribute which already exists is given next is explained using an example. If the XML data shown in drawing 10 are given when the integrated tree structure shown in drawing 7 is already memorized, CPU23 will memorize the given XML data to container box 26c, not breaking down the attribute structure (drawing 8 step S1). This processing is concerned with whether it exists or not and is [that there is nothing] the same. Next, CPU23 extracts the tree structure shown in drawing 11 from these XML data (step S3). CPU23 initializes the processing number i (step S5), and makes the i-th attribute an attention attribute (step S7). It judges whether CPU23 already exists in an integrated tree structure about an attention attribute (step S9). In this case, since an attribute "an identifier" already exists in the integrated tree structure of drawing 4 , it progresses to step S13 and the start address in said container box of the data "Goro Sato" belonging to the attribute concerned is memorized with the block id concerned for the attribute "an identifier" of said integrated tree structure.

[0042] CPU23 performs same processing about an attribute "affiliation", the "section", and a "section." It adds to an integrated tree structure as a new attribute noting that the target attribute does not exist in drawing 8 step S9, since it next does not exist in the integrated tree structure shown in drawing 7 about an attribute "charge" (step S11).

[0043] It is similarly added to an integrated tree structure about an attribute "the telephone number (carrying)." Thereby, as an integrated tree structure is shown in drawing 12 , an attribute "charge" and "the telephone number (carrying)" are added.

[0044] (3.2) Explain data retrieval processing using the flow chart of drawing 13 about

data retrieval processing. If a retrieval instruction is given, CPU23 will acquire the structure of an integrated tree structure from index 26for retrieval i (step S21). Here, it explains as what the retrieval instruction which extracts the identifier of the personnel who belong to the 1st section of a marketing department as a retrieval instruction was given.

[0045] CPU23 makes said integrated tree structure reference, and the attribute for retrieval judges that it is a low-ranking attribute "a section" to the pan of the attribute "the section" of the low order of an attribute "affiliation", and reads the index data (Block id and start address) memorized by the attribute "a section" concerned (step S23). if it is the example shown in drawing 7 -- "block id=0, the address 0008", "block id=1 and the address 0020", and "block id=3 and the address 0035" ... is read. CPU23 searches container box 26c, and reads live data from these index data (step S25).

[0046] CPU23 chooses what agrees on retrieval conditions among the read live data (step S27). What is necessary is to specify Block id from selected data and just to extract from a container box. What is necessary is to acquire the start address about the block the block id of the index data memorized by said attribute "an identifier" of the index data for retrieval and said whose specified block id specifically correspond, and just to make it extract from a container box.

[0047] Thus, in this operation gestalt, the structure of each of said attribute item is memorized and CPU23 searches the data which made structure of each attribute item reference and were memorized by storage container section 26c. Therefore, even when it memorizes not breaking down DS about XML data, high-speed search becomes possible.

[0048] Moreover, since the integrated tree structure is memorized, retrieval becomes possible even if even the attribute of the high order of a certain attribute is not given as retrieval conditions.

[0049] In addition, as a retrieval instruction, when the attribute of the high order is given with the attribute for retrieval, processing of step S21 is unnecessary.

[0050] In addition, what is necessary is just to carry out processing contrary to an addition, in carrying out data deletion. Namely, what is necessary is just to delete the data which correspond from the index for retrieval, while deleting from a container box.

[0051] Moreover, what is necessary is to cancel the data which once correspond and just to newly give the XML data to which this attribute was added, in adding a new attribute about the already memorized block data. However, it ignores and you may make it add only the attribute which should be added to block data and the index for

retrieval about the attribute which already exists besides this, and its value. For example, what is necessary is just to add an attribute "the telephone number (carrying)" to the index for retrieval, while extracting a different attribute and adding to the block data concerned, in CPU's23 searching a container box if the XML data shown in drawing 14 are given when adding the attribute "a cellular phone" which is a new attribute, and judging whether the same block exists and existing about personnel "Ichiro Yamada."

[0052] Thus, in this operation gestalt, while memorizing not the record format that is the form where it was suitable for the conventional retrieval in the data format of a database but the structure of XML data as it is, the index for retrieval which shows a data's existence location for every attribute is adopted. Therefore, a flexible database management system with free modification of data format etc. can be offered. Moreover, the database management system suitable for the data control of the data constellation from which the attribute of each item has a tree structure can be offered.

[0053] In addition, in this operation gestalt, although the case where XML data were used as data structurally specified in the attribute was explained, if it is data structurally specified in the attribute, it is arbitrary to the data format.

[0054] In addition, an operating system (OS) program may be made to perform a part among said each program. Namely, it may carry out by the program independent or may be made to realize by assigning an operating system.

[0055] In addition, in this operation gestalt, in order to realize the function shown in drawing 1 , CPU23 was used and software has realized this. However, a part may be realized by hardware. Moreover, about the habitat segregation of hardware and software, the combination of arbitration is possible.

[Translation done.]

* NOTICES *

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] Especially this invention relates to improvement in the degree of freedom of DS about a database management system.

[Translation done.]

* NOTICES *

JPO and NCIPI are not responsible for any
damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] Generally, in the database management system, it has two or more records and each record consists of two or more fields. The DS (data type, data length, etc.) of this field is defined beforehand. Thus, by defining the data of each field beforehand, high-speed search becomes possible.

[0003] By the way, XML data attract attention as structural description mold data today. Since these XML data can make an attribute add to the data itself, the given data can carry out automatic extracting of whether it is data to which field. For example, if it is XML data shown in drawing 3 , data "Ichiro Yamada" can be memorized in the field "an identifier", and a computer can make the field "an executive" memorize data "a section chief" by referring to the tag of each item.

[Translation done.]

* NOTICES *
JPO and NCIPI are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] However, there were the following problems in the above-mentioned database management system. When each record is made into a fixed length, the number of the fields and field length for every record turn into a fixed length. Therefore, DS must be reconstructed when newly changing an addition and field length of the field. That is, even if XML data including a new attribute are given, the data of the attribute concerned are disregarded. The amount of data which a database management system manages has many huge things, for example, it becomes very complicated processing to change about all DS for the new field which only one record of the 1 million records has.

[0005] Moreover, even when data do not exist in the field concerned, the data input of the data (null etc.) set by the DS concerned must be carried out, and a useless field occurs.

[0006] Although making each record into variable length is also considered in order to solve this problem, it is necessary to define the number of the fields of each record beforehand even in this case only by the ability to make field length of each field into variable length. Therefore, the addition of the new field cannot be performed.

[0007] That is, in the conventional database management system, it had not become the structure of utilizing enough the merit of the XML data that an attribute can be added to the data itself. Moreover, although XML data could describe the relational structure of each item using said tag, in the conventional database management system, this relational structure was not memorizable as it was.

[0008] This invention solves the above-mentioned problem and it aims at offering the flexible database management system which can fully employ that property efficiently

about the description mold data of an attribute good addition mold.

[Translation done.]

* NOTICES *
JPO and NCIPI are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] 1) The database management system concerning this invention The XML block data with which data are memorized about two or more attribute items, and each attribute item consisted of tree structures While it is the database management system managed using a computer, and memorizing 1 aforementioned XML block data for every XML block data to the 1st storage region, with the structure concerned held the index data for retrieval which memorize data's belonging to each attribute item of each XML block data existence positional information collectively for every attribute -- the 2nd storage region -- memorizing -- 2, if new XML block data is given While carrying out the additional storage of performing the following processings and the A aforementioned XML block data in said 1st storage region, with the tree structure concerned held B) about the attribute item which is in agreement with the attribute item of said index data for retrieval among the attribute items of said newly given XML block data About the attribute item which adds data's of attribute item concerned existence positional information to said 2nd storage region, and does not exist in the attribute item of said index data for retrieval While adding to said 2nd storage region as a new attribute item, data's of attribute item concerned existence positional information is added to said 2nd storage region. Therefore, the XML block data with which each attribute item consisted of tree structures can be memorized, without breaking down the structure concerned.

Furthermore, additional XML block data can be memorized altogether, without breaking down the structure concerned. Moreover, since said index data for retrieval are gathered for every attribute, the data of a certain attribute can be searched at a high speed. The database management system which can utilize by this the property of the XML data that an attribute can be added to each data can be offered.

[0010] 2) In the database management system concerning this invention 1) It is the data constellation data were remembered to be about two or more attribute items. A data constellation storage means to memorize two or more data constellations as which each attribute item is determined structurally, without breaking down the structure concerned, 2) An auxiliary-data storage means to memorize the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute, 3) If the data constellation data were remembered to be about two or more attribute items is newly given, while carrying out additional storage, for said data storage means About the attribute item which is in agreement with the attribute item of said retrieval auxiliary data among the attribute items of said newly given data constellation Data's of attribute item concerned existence positional information was added, and about the attribute item which does not exist in the attribute item of said retrieval auxiliary data, while adding as a new attribute item, it has an additional data group processing means to add data's of attribute item concerned existence positional information.

[0011] Therefore, the data constellation as which each attribute item is determined structurally can be memorized, without breaking down the structure concerned. Furthermore, an additional data constellation can be memorized altogether, without breaking down the structure concerned. Moreover, since said retrieval auxiliary data is gathered for every attribute, the data of a certain attribute can be searched at a high speed. The database management system which can utilize by this the property of the attribute addition mold data that an attribute can be added to each data can be offered.

[0012] 3) In the database management system concerning this invention, the structure of each of said attribute item is a tree structure. Therefore, the data constellation which has the attribute item of a tree structure is manageable as a database.

[0013] 4) The database management system concerning this invention 1) It is the data constellation data were remembered to be about two or more attribute items. A data constellation storage means to memorize two or more data constellations as which

each attribute item is determined structurally, without breaking down the structure concerned, 2) If the retrieval instruction including an auxiliary-data storage means to memorize the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute, and the attribute item which carries out 3 retrieval is given The retrieval auxiliary data memorized by said auxiliary-data storage means was made reference, and it has a retrieval means to search the data memorized by the data storage means. Thus, even when the data constellation structurally constituted by searching the desired data with which each attribute item exists in the data constellation defined structurally using said retrieval auxiliary data is memorized as it was, data retrieval becomes possible at a high speed.

[0014] 5) In the database management system concerning this invention, said auxiliary-data storage means has memorized the structure of each of said attribute item, and said retrieval means searches the data which made structure of each attribute item reference and were memorized by the data storage means. Therefore, even if only the attribute item searched as a retrieval instruction is given, high-speed search becomes possible.

[0015] 6) In the data-logging medium which recorded the data concerning this invention A) It is the data constellation data were remembered to be about two or more attribute items. It has the auxiliary-data storage region which memorizes collectively the retrieval auxiliary data which memorizes data's which belong data constellation's as which each attribute item's is determined structurally to data constellation storage region and each attribute item's of data constellation of B aforementioned plurality which are memorized without breaking down structure's concerned existence positional information for every attribute. Therefore, even if it memorizes two or more data constellations as which each attribute item is determined structurally, without breaking down the structure concerned, they can be searched at high speed using said retrieval auxiliary data.

[0016] 7) In the data retrieval approach of searching data using the computer concerning this invention 1) It is the data constellation data were remembered to be about two or more attribute items. The data constellation as which each attribute item is determined structurally, without breaking down the structure concerned Plurality, The retrieval auxiliary data which memorizes to the 1st storage region and memorizes data's belonging to each attribute item of two or more 2 aforementioned data constellations existence positional information collectively for every attribute is memorized to the 2nd storage region. 3) If the retrieval instruction including the

attribute item to search is given, said retrieval auxiliary data will be made reference and the data constellation memorized in said 1st storage region will be searched. Thus, data retrieval becomes possible from the data constellation constituted structurally at a high speed by searching the desired data with which each attribute item exists in the data constellation defined structurally using said retrieval auxiliary data.

[0017] 8) In the data control approach using the computer concerning this invention Are the data constellation data were remembered to be about two or more attribute items, and the data constellation as which each attribute item is determined structurally The retrieval auxiliary data which memorizes more than one to the 1st storage region, without breaking down the structure concerned, and memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute is memorized to the 2nd storage region. If the data constellation data were remembered to be about two or more attribute items is newly given, while carrying out additional storage in said 1st storage region About the attribute item which is in agreement with the attribute item of said retrieval auxiliary data among the attribute items of said newly given data constellation Data's of attribute item concerned existence positional information is added, and about the attribute item which does not exist in the attribute item of said retrieval auxiliary data, while adding as a ******** attribute item, data's of attribute item concerned existence positional information is added to said 2nd storage region. Therefore, two or more data constellations as which each attribute item is determined structurally can be memorized, without breaking down the structure concerned. Moreover, an additional data constellation can be memorized altogether, without breaking down the structure concerned.

[0018] Although XML data were used for "the description mold data of an attribute good addition mold" with the operation gestalt, as long as it is data which are not limited to this but can add an attribute, they may be what kind of thing. Moreover, you may be except the method specified with a tag like [technique / of attribute addition] XML data.

[0019]

[Embodiment of the Invention] 1. Explain 1 operation gestalt of explanation this invention of a functional block diagram based on a drawing. The database management system 1 shown in drawing 1 is equipped with the data constellation storage means 5, the auxiliary-data storage means 7, the retrieval means 3, and the additional data group processing means 9.

[0020] The data constellation storage means 5 is the data constellation data were

remembered to be about two or more attribute items, and they are memorized, without breaking down the structure concerned for the data constellation as which each attribute item is determined structurally. [two or more] The auxiliary-data storage means 7 memorizes the retrieval auxiliary data which memorizes data's belonging to each attribute item of two or more of said data constellations existence positional information collectively for every attribute. If the retrieval instruction including the attribute item to search is given, the retrieval means 3 will make reference the retrieval auxiliary data memorized by the auxiliary-data storage means 7, and will search the data memorized by the data storage means 5.

[0021] Therefore, data retrieval becomes possible from the data constellation constituted structurally at a high speed. Moreover, since the retrieval auxiliary data was gathered for every attribute and said existence positional information is memorized, high-speed search becomes more possible.

[0022] If the data constellation data were remembered to be about two or more attribute items is newly given, while the additional data group processing means 9 carries out additional storage, for the data storage means 5 About the attribute item which is in agreement with the attribute item of said retrieval auxiliary data among the attribute items of said newly given data constellation Data's of attribute item concerned existence positional information is added, and about the attribute item which does not exist in the attribute item of said retrieval auxiliary data, while adding as a ******** attribute item, the additional storage of the data's of attribute item concerned existence positional information is carried out at said auxiliary-data storage means 7. Therefore, two or more data constellations as which each attribute item is determined structurally can be memorized as they are, without breaking down the structure concerned. Moreover, an additional data constellation can be memorized altogether, without breaking down the structure concerned.

[0023] 2. An example of a hardware configuration which realized the database management system 1 shown in drawing 1 to hardware configuration drawing 2 using CPU is shown. The database management system 1 is equipped with CPU23, memory 27, a hard disk 26, FDD25, the monitor 32, the keyboard 28, the mouse 31, and the bus line 29.

[0024] CPU23 controls each part through a bus line 29 according to the program memorized by the hard disk 26. This program is read from flexible disk 25a the program was remembered to be through FDD25, and is installed on a hard disk 26. In addition, you may make it make the program of CD-ROM, an IC card, etc. install on a hard disk from the computer-readable record medium unified ontically in addition to a

flexible disk. Furthermore, you may make it download using a communication line.
[0025] He is trying to make a computer execute indirectly the program stored in the flexible disk in this operation gestalt by making a program install on a hard disk 26 from a flexible disk. However, it may be made to perform the program stored in the flexible disk directly from FDD25, without being limited to this. In addition, by computer, the thing in which immediate execution is possible only by [as it is] installing as a program which can be executed combines with a thing (for example, that by which the data compression is carried out is thawed) to be changed, and the module part of further others, and of course once contains in other gestalten etc. what can be performed.
[0026] In a hard disk 26, a control program and an operating system (OS) are memorized. As an operating system, WindowsNT (trademark) by Microsoft [Corp.] Corp. is employable, for example.
[0027] Container box 26c and index data 26i for retrieval are further memorized by the hard disk 26. Two or more storage of the XML data with which the attribute was added to each data as shown in drawing 3 is carried out at container box 26c, not breaking down the attribute-structure relation. When it explains concretely, the XML data shown in drawing 3 A are expressed with a tree structure as shows each attribute-structure relation to drawing 4 . In order to memorize this tree structure as it is, as shown in drawing 5 , the pointer in which it is shown to which field the live data about each attribute are memorized is memorized. For example, about the attribute "an identifier", the pointer P1 in which the start address of live data "Ichiro Yamada" is shown is memorized. Moreover, in this operation gestalt, since each attribute-structure relation is expressed with the tree structure The attribute "the section" belongs to the attribute "affiliation" further, and the live data are memorized by the pointer P5. Furthermore, the attribute "a section" is memorized in the field to which it points with a pointer P5 in this case, and the live data of an attribute "a section" are memorized to the field to which it points with a pointer P6. In this operation gestalt, the data constellation of the XML data of 1 is called block data. Two or more storage of this block data is carried out at container box 26c.
[0028] Index data 26for retrieval i is explained using drawing 7 . Index data 26i for retrieval is index data for searching two or more block data memorized by container box 26c, and is memorized by the integrated tree structure which unified the tree structure of each attribute of two or more block data memorized by container box 26c. Furthermore, they are collectively memorized by Container id and the start address (index information) of live data for every attribute about the block data which has

[attributes / of an integrated tree structure / all] data about the attribute concerned.
[0029] An integrated tree structure is explained. An integrated tree structure means the tree structure with which the tree structure of all block data was unified by having performed the OR operation about the tree structure of all block data. For example, as shown in drawing 6 , when there are a tree structure shown in drawing 6 A, a tree structure shown in drawing 6 B, and a tree structure shown in drawing 6 C, drawing 6 D serves as an integrated tree structure.
[0030] Block id and the start address (index information) of the live data of block data which have data for the attribute concerned are memorized by each attribute of this integrated tree structure. for example, the address 0001 in the block id with which the live data "Ichiro Yamada" of an attribute [in / in an attribute "an identifier" / in an integrated tree structure / the block data of block id=0] "an identifier" are stored, and a container box and the address 0021 in block id=1 in which the live data "Jiro Tanaka" of the attribute "an identifier" in the block data of block id=1 are stored, and a container box .... as -- the data for retrieval are memorized. About how to use this address for retrieval, it mentions later.
[0031] In addition, since it is an integrated tree structure, it may not have [block data / a certain] data about the attribute concerned. For example, since the data of block id=2 do not exist in an attribute "a section" when an attribute "a section" does not exist [the data of block id=2] in an attribute "affiliation" with an attribute "the Administrative Division", the index is not memorized.
[0032] Thus, the structure of each attribute item is a tree structure, and as for the block data memorized by container box 26c, they are collectively memorized by index data for data to search the existing block data for every attribute about each attribute.
[0033] Various kinds of results of an operation etc. are memorized by memory 27.
[0034] A keyboard 28 and a mouse 31 are input means to input various kinds of data or an instruction. A retrieval result etc. is displayed on a monitor 32.
[0035] 3. Explain flow chart (3.1) data addition storage processing using drawing 8 about the data storage processing to a hard disk 26. While block data with a data addition new to container box 26c is added, index data are added to index 26i for retrieval. Here, when the integrated tree structure shown in drawing 4 was already memorized, the XML data shown in drawing 9 A should be given. CPU23 memorizes the given XML data to container box 26c, not breaking down the attribute structure (step S1). Thereby, one block data is added.
[0036] CPU23 extracts the tree structure shown in drawing 9 B from these XML data (step S3). This is realizable by extracting the relational structure of the tag of XML

data. For example, it turns out that a tag "an identifier", "affiliation", an "executive", and the "telephone number" are located in the low order of a tag "personnel" from drawing 9 A, the tag "the section" is further located in the low order of a tag "an executive", and the tag "a section" is further located in the low order of a tag "the section." Thereby, as shown in drawing 9 B, the tree structure of each attribute can be extracted.

[0037] CPU23 initializes the processing number i (step S5), and makes the i-th attribute an attention attribute (step S7). It judges whether CPU23 already exists in an integrated tree structure about an attention attribute (step S9). In this case, since an attribute "an identifier" already exists in the integrated tree structure of drawing 7 , it progresses to step S13 and the start address in said container box of the data "Taro Suzuki" belonging to the attribute concerned is memorized with the block id concerned for the attribute "an identifier" of said integrated tree structure.

[0038] CPU23 judges whether the attention attribute which is not examined exists (step S15), when it exists, increments the processing number i (step S17), and performs processing not more than step S7. In this case, since it already exists altogether about the remaining attribute "affiliation", the "section", a "section", an "executive", and the "telephone number", only new address data are memorized similarly (step S13), and processing will be ended if all items are examined.

[0039] In addition, what is necessary is just to judge that whether an attribute exists or not does not already exist when the attributes of the high order differ. Namely, what is necessary is just to judge that it already exists in full coincidence. For example, what is necessary is just to judge that it is a different attribute, when an attribute "a section" exists under affiliation although an attribute "a section" is located in the lower part of an attribute "the section" in this case.

[0040] Thus, without breaking down the attribute-structure relation, new XML data are memorized and additional storage is further carried out at index 26i for retrieval.

[0041] The case where the attribute which is not an attribute which already exists is given next is explained using an example. If the XML data shown in drawing 10 are given when the integrated tree structure shown in drawing 7 is already memorized, CPU23 will memorize the given XML data to container box 26c, not breaking down the attribute structure (drawing 8 step S1). This processing is concerned with whether it exists or not and is [that there is nothing] the same. Next, CPU23 extracts the tree structure shown in drawing 11 from these XML data (step S3). CPU23 initializes the processing number i (step S5), and makes the i-th attribute an attention attribute (step S7). It judges whether CPU23 already exists in an integrated tree structure

about an attention attribute (step S9). In this case, since an attribute "an identifier" already exists in the integrated tree structure of drawing 4 , it progresses to step S13 and the start address in said container box of the data "Goro Sato" belonging to the attribute concerned is memorized with the block id concerned for the attribute "an identifier" of said integrated tree structure.

[0042] CPU23 performs same processing about an attribute "affiliation", the "section", and a "section." It adds to an integrated tree structure as a new attribute noting that the target attribute does not exist in drawing 8 step S9, since it next does not exist in the integrated tree structure shown in drawing 7 about an attribute "charge" (step S11).

[0043] It is similarly added to an integrated tree structure about an attribute "the telephone number (carrying)." Thereby, as an integrated tree structure is shown in drawing 12 , an attribute "charge" and "the telephone number (carrying)" are added.

[0044] (3.2) Explain data retrieval processing using the flow chart of drawing 13 about data retrieval processing. If a retrieval instruction is given, CPU23 will acquire the structure of an integrated tree structure from index 26for retrieval i (step S21). Here, it explains as what the retrieval instruction which extracts the identifier of the personnel who belong to the 1st section of a marketing department as a retrieval instruction was given.

[0045] CPU23 makes said integrated tree structure reference, and the attribute for retrieval judges that it is a low-ranking attribute "a section" to the pan of the attribute "the section" of the low order of an attribute "affiliation", and reads the index data (Block id and start address) memorized by the attribute "a section" concerned (step S23). if it is the example shown in drawing 7 -- "block id=0, the address 0008", "block id=1 and the address 0020", and "block id=3 and the address 0035" ... is read. CPU23 searches container box 26c, and reads live data from these index data (step S25).

[0046] CPU23 chooses what agrees on retrieval conditions among the read live data (step S27). What is necessary is to specify Block id from selected data and just to extract from a container box. What is necessary is to acquire the start address about the block the block id of the index data memorized by said attribute "an identifier" of the index data for retrieval and said whose specified block id specifically correspond, and just to make it extract from a container box.

[0047] Thus, in this operation gestalt, the structure of each of said attribute item is memorized and CPU23 searches the data which made structure of each attribute item reference and were memorized by storage container section 26c. Therefore, even

when it memorizes not breaking down DS about XML data, high-speed search becomes possible.

[0048] Moreover, since the integrated tree structure is memorized, retrieval becomes possible even if even the attribute of the high order of a certain attribute is not given as retrieval conditions.

[0049] In addition, as a retrieval instruction, when the attribute of the high order is given with the attribute for retrieval, processing of step S21 is unnecessary.

[0050] In addition, what is necessary is just to carry out processing contrary to an addition, in carrying out data deletion. Namely, what is necessary is just to delete the data which correspond from the index for retrieval, while deleting from a container box.

[0051] Moreover, what is necessary is to cancel the data which once correspond and just to newly give the XML data to which this attribute was added, in adding a new attribute about the already memorized block data. However, it ignores and you may make it add only the attribute which should be added to block data and the index for retrieval about the attribute which already exists besides this, and its value. For example, what is necessary is just to add an attribute "the telephone number (carrying)" to the index for retrieval, while extracting a different attribute and adding to the block data concerned, in CPU's23 searching a container box if the XML data shown in drawing 14 are given when adding the attribute "a cellular phone" which is a new attribute, and judging whether the same block exists and existing about personnel "Ichiro Yamada."

[0052] Thus, in this operation gestalt, while memorizing not the record format that is the form where it was suitable for the conventional retrieval in the data format of a database but the structure of XML data as it is, the index for retrieval which shows a data's existence location for every attribute is adopted. Therefore, a flexible database management system with free modification of data format etc. can be offered. Moreover, the database management system suitable for the data control of the data constellation from which the attribute of each item has a tree structure can be offered.

[0053] In addition, in this operation gestalt, although the case where XML data were used as data structurally specified in the attribute was explained, if it is data structurally specified in the attribute, it is arbitrary to the data format.

[0054] In addition, an operating system (OS) program may be made to perform a part among said each program. Namely, it may carry out by the program independent or may be made to realize by assigning an operating system.

[0055] In addition, in this operation gestalt, in order to realize the function shown in

drawing 1 , CPU23 was used and software has realized this. However, a part may be realized by hardware. Moreover, about the habitat segregation of hardware and software, the combination of arbitration is possible.

[Translation done.]

* NOTICES *
JPO and NCIPI are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the functional block diagram of the database management system 1 concerning this invention.
[Drawing 2] It is drawing showing an example of the hardware configuration which constituted the database management system 1 using CPU.
[Drawing 3] It is drawing showing an example of the XML data inputted.
[Drawing 4] it can set to the XML data of drawing 3 -- it is shown an attribute structure (tree structure).
[Drawing 5] The relation of each data at the time of memorizing the XML data of drawing 3 as block data is shown.
[Drawing 6] It is drawing showing an integration tree structure.
[Drawing 7] It is the DS of the index data for retrieval.
[Drawing 8] It is the flow chart of data storage processing.
[Drawing 9] It is drawing showing an example of the XML data by which an additional input is carried out.
[Drawing 10] It is drawing showing an example of the XML data by which an additional input is carried out.

[Drawing 11] It is drawing showing the attribute structure of the XML data of drawing 10 .

[Drawing 12] It is the DS of the index data for retrieval after an addition.

[Drawing 13] It is the flow chart of data retrieval processing.

[Drawing 14] It is drawing showing an example of the XML data by which an additional input is carried out.

[Description of Notations]

1 ... Database management system

23 .... CPU

26 .... Hard disk

26i .... Index for retrieval

26c .... Container box

[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2001－195406
(P2001－195406A)
(43)公開日　平成13年７月19日(2001.7.19)

(51)Int.Cl.[7]		識別記号	FI			テーマコード(参考)
Ｇ０６Ｆ	17/30		Ｇ０６Ｆ	12/00	５２０Ｊ	５Ｂ０７５
	12/00	５２０		15/401	３３０Ｚ	５Ｂ０８２
				15/40	３４０	
				15/413	３１０Ａ	

審査請求　有　　請求項の数９　ＯＬ　（全 12 頁）

(21)出願番号　特願2000－867(P2000－867)

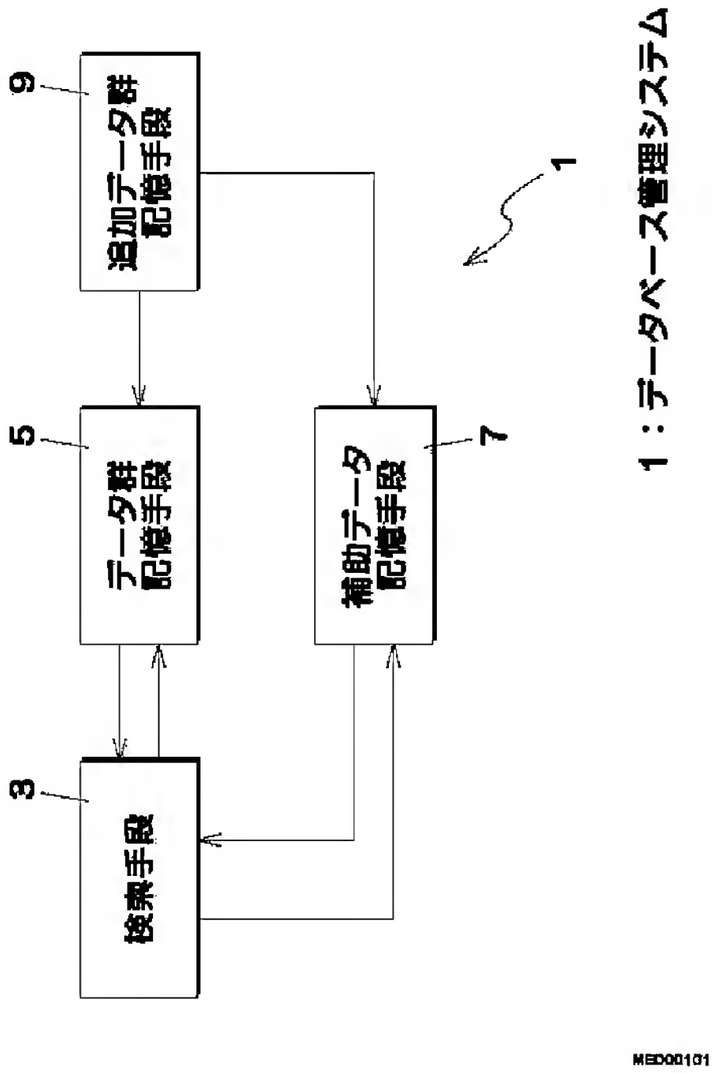
(22)出願日　平成12年１月６日(2000.1.6)

(71)出願人　596146544
株式会社メディアフュージョン
兵庫県尼崎市道意町７丁目１番３号
(71)出願人　500016051
織田　美樹男
兵庫県西宮市安井町５－４
(72)発明者　榊原　淳
兵庫県神戸市垂水区小束山１－２－６－101
(72)発明者　織田　美樹男
兵庫県西宮市安井町５－４
(74)代理人　100092956
弁理士　古谷　栄男　（外２名）

最終頁に続く

(54)【発明の名称】　データベース管理システム

(57)【要約】
【課題】　ＸＭＬブロックデータをそのまま記憶できるデータベース管理システムを提供する。
【解決手段】　データ群記憶手段５は、複数の属性項目についてデータが記憶され、かつ各属性項目が構造的に定められたブロックデータを、複数記憶する。補助データ記憶手段７は、複数のブロックデータの各属性項目に属するデータの存在位置情報を含む検索補助データを記憶する。検索手段３は、検索する属性項目を含む検索命令が与えられると、補助データ記憶手段７に記憶された検索補助データを参照にして、データ記憶手段５に記憶されたデータを検索する。したがって、各属性項目が構造的に定められているブロックデータを、そのまま、複数記憶しても、高速にデータ検索が可能となる。



1

【特許請求の範囲】
【請求項１】複数の属性項目についてデータが記憶されており、かつ、各属性項目がツリー構造で構成されたＸＭＬブロックデータを、コンピュータを用いて管理するデータベース管理システムであって、
前記ＸＭＬブロックデータを当該構造を保持したまま、第１の記憶領域にＸＭＬブロックデータごとに記憶するとともに、各ＸＭＬブロックデータの各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索用インデックスデータを、第２の記憶領域に記憶しておき、
新たなＸＭＬブロックデータが与えられると、以下の処理を行うこと、
A)前記ＸＭＬブロックデータを当該ツリー構造を保持したまま、前記第１の記憶領域に追加記憶するとともに、
B)前記新たに与えられたＸＭＬブロックデータの属性項目のうち、前記検索用インデックスデータの属性項目と一致する属性項目については、当該属性項目のデータの存在位置情報を前記第２の記憶領域に追加し、前記検索用インデックスデータの属性項目に存在しない属性項目については、新たな属性項目として前記第２の記憶領域に追加するとともに、当該属性項目のデータの存在位置情報を前記第２の記憶領域に追加する、
を特徴とするデータベース管理システム。
【請求項２】複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶するデータ群記憶手段、
前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを記憶する補助データ記憶手段、
複数の属性項目についてデータが記憶されたデータ群が新たに与えられると、前記データ記憶手段に追加記憶するとともに、前記新たに与えられたデータ群の属性項目のうち、前記検索補助データの属性項目と一致する属性項目については、当該属性項目のデータの存在位置情報を追加し、前記検索補助データの属性項目に存在しない属性項目については、新たな属性項目として追加するとともに、当該属性項目のデータの存在位置情報を追加する追加データ群処理手段、
を備えたデータベース管理システム。
【請求項３】請求項２のデータベース管理システムにおいて、
前記各属性項目の構造はツリー構造であること、
を特徴とするもの。
【請求項４】複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶するデータ群記憶手段、
前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを記憶する補助データ記憶手段、
検索する属性項目を含む検索命令が与えられると、前記補助データ記憶手段に記憶された検索補助データを参照にして、データ記憶手段に記憶されたデータを検索する検索手段、
を備えたデータベース管理システム。

2

【請求項５】請求項４のデータベース管理システムにおいて、
前記補助データ記憶手段は、前記各属性項目の構造を記憶しており、
前記検索手段は、各属性項目の構造を参照にしてデータ記憶手段に記憶されたデータを検索すること、
を特徴とするもの。
【請求項６】Ａ）複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶するデータ群記憶領域、
Ｂ）前記複数のデータ群の各属性項目に属するデータの存在位置情報を記憶する検索補助データを各属性毎にまとめて記憶する補助データ記憶領域、を有するデータ記録媒体。
【請求項７】コンピュータを用いてデータを検索するデータ検索方法であって、
複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数、第１の記憶領域に記憶しておき、
前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを第２の記憶領域に記憶しておき、
検索する属性項目を含む検索命令が与えられると、前記検索補助データを参照にして、前記第１の記憶領域に記憶されたデータ群を検索すること、
を特徴とするデータ検索方法。
【請求項８】コンピュータを用いたデータ管理方法であって、
複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数第１の記憶領域に記憶しておき、
前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを第２の記憶領域に記憶しておき、
複数の属性項目についてデータが記憶されたデータ群が新たに与えられると、前記第１の記憶領域に追加記憶するとともに、前記新たに与えられたデータ群の属性項目のうち、前記検索補助データの属性項目と一致する属性項目については、当該属性項目のデータの存在位置情報を追加し、前記検索補助データの属性項目に存在しない

属性項目については、あたらな属性項目として追加するとともに、当該属性項目のデータの存在位置情報を前記第２の記憶領域に追加すること、
を特徴とするデータ管理方法。
【請求項９】あらかじめ第１の記憶領域に、複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶するとともに、第２の記憶領域に前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを記憶したコンピュータをデータベース管理装置として実現するプログラムを記憶した記憶媒体であって、検索する属性項目を含む検索命令が与えられると、前記検索補助データを参照にして、前記第１の記憶領域に記憶されたデータ群を検索する処理を前記コンピュータに実行させるためのプログラムを記憶した記憶媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、データベース管理システムに関し、特に、データ構造の自由度の向上に関する。
【０００２】
【従来技術】一般に、データベース管理システムにおいては、複数のレコードを有しており、各レコードは複数のフィールドから構成されている。かかるフィールドのデータ構造（データ型、データ長など）は予め定められている。このように各フィールドのデータをあらかじめ定義しておくことにより、高速検索が可能となる。
【０００３】ところで、今日、構造記述型データとしてＸＭＬデータが注目されている。かかるＸＭＬデータはデータ自体に属性を付加させることができるので、与えられたデータがどのフィールドにデータであるかを自動抽出できる。例えば、図３に示すＸＭＬデータであれば、コンピュータは、各項目のタグを参照することにより、データ「山田一郎」をフィールド「名前」に、データ「課長」をフィールド「役職」に記憶させることができる。
【０００４】
【発明が解決しようとする課題】しかしながら、上記データベース管理システムにおいては、以下のような問題があった。各レコードを固定長とした場合、各レコード毎のフィールド数およびフィールド長は固定長となる。したがって、新たにフィールドの追加やフィールド長を変更する場合には、データ構造を再構築しなければならない。すなわち、新たな属性を含むＸＭＬデータが与えられても、当該属性のデータは無視される。データベース管理システムが管理するデータ量は膨大なものが多く、例えば１００万レコードのうちの１レコードだけが有する新たなフィールドの為に、全データ構造について変更するのは非常に煩雑な処理となる。
【０００５】また、当該フィールドにデータが存在しない場合でも、当該データ構造にあわせたデータ（nullなど）をデータ入力しておかなければならず、無駄な領域が発生する。
【０００６】かかる問題を解決するために、各レコードを可変長とすることも考えられるが、この場合でも、各フィールドのフィールド長を可変長とすることができるだけで、各レコードのフィールド数はあらかじめ定める必要がある。したがって、新たなフィールドの追加はできない。
【０００７】すなわち、従来のデータベース管理システムでは、データ自身に属性を付加できるというＸＭＬデータのメリットを十分活用する構造となっていなかった。また、ＸＭＬデータは前記タグを用いて、各項目の関係構造を記述することができるが、従来のデータベース管理システムではかかる関係構造をそのまま記憶することができなかった。
【０００８】この発明は上記問題を解決し、属性可付加型の記述型データについて、その特性を十分に生かすことのできる柔軟なデータベース管理システムを提供することを目的とする。
【０００９】
【課題を解決するための手段】１）本発明にかかるデータベース管理システムは、複数の属性項目についてデータが記憶されており、かつ、各属性項目がツリー構造で構成されたＸＭＬブロックデータを、コンピュータを用いて管理するデータベース管理システムであって、1)前記ＸＭＬブロックデータを当該構造を保持したまま、第１の記憶領域にＸＭＬブロックデータごとに記憶するとともに、各ＸＭＬブロックデータの各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索用インデックスデータを、第２の記憶領域に記憶しておき、2)新たなＸＭＬブロックデータが与えられると、以下の処理を行うこと、A)前記ＸＭＬブロックデータを当該ツリー構造を保持したまま、前記第１の記憶領域に追加記憶するとともに、B)前記新たに与えられたＸＭＬブロックデータの属性項目のうち、前記検索用インデックスデータの属性項目と一致する属性項目については、当該属性項目のデータの存在位置情報を前記第２の記憶領域に追加し、前記検索用インデックスデータの属性項目に存在しない属性項目については、新たな属性項目として前記第２の記憶領域に追加するとともに、当該属性項目のデータの存在位置情報を前記第２の記憶領域に追加する。したがって、各属性項目がツリー構造で構成されたＸＭＬブロックデータを、当該構造を崩すことなく記憶できる。さらに、追加のＸＭＬブロックデータを当該構造を崩すことなくすべて記憶することができる。また、前記検索用インデックスデータは各属性毎にまとめられているので、ある属性のデータを高速に検索できる。これにより、各データに属性を付加できるというＸ

ＭＬデータの特性を活用できるデータベース管理システムを提供することができる。

【００１０】２）本発明にかかるデータベース管理システムにおいては、1)複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶するデータ群記憶手段、2)前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを記憶する補助データ記憶手段、3)複数の属性項目についてデータが記憶されたデータ群が新たに与えられると、前記データ記憶手段に追加記憶するとともに、前記新たに与えられたデータ群の属性項目のうち、前記検索補助データの属性項目と一致する属性項目については、当該属性項目のデータの存在位置情報を追加し、前記検索補助データの属性項目に存在しない属性項目については、新たな属性項目として追加するとともに、当該属性項目のデータの存在位置情報を追加する追加データ群処理手段を備えている。

【００１１】したがって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく記憶できる。さらに、追加のデータ群を当該構造を崩すことなくすべて記憶することができる。また、前記検索補助データは各属性毎にまとめられているので、ある属性のデータを高速に検索できる。これにより、各データに属性を付加できるという属性付加型データの特性を活用できるデータベース管理システムを提供することができる。

【００１２】３）本発明にかかるデータベース管理システムにおいては、前記各属性項目の構造はツリー構造である。したがって、ツリー構造の属性項目を有するデータ群をデータベースとして管理することができる。

【００１３】４）本発明にかかるデータベース管理システムは、1)複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶するデータ群記憶手段、2)前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを記憶する補助データ記憶手段、3)検索する属性項目を含む検索命令が与えられると、前記補助データ記憶手段に記憶された検索補助データを参照にして、データ記憶手段に記憶されたデータを検索する検索手段を備えている。このように、各属性項目が構造的に定められているデータ群に存在する所望のデータを、前記検索補助データを用いて検索することにより、構造的に構成されたデータ群をそのまま記憶した場合でも、高速にデータ検索が可能となる。

【００１４】５）本発明にかかるデータベース管理システムにおいては、前記補助データ記憶手段は、前記各属性項目の構造を記憶しており、前記検索手段は、各属性項目の構造を参照にしてデータ記憶手段に記憶されたデータを検索する。したがって、検索命令として検索する

属性項目だけが与えられても、高速検索が可能となる。

【００１５】６）本発明にかかるデータを記録したデータ記録媒体においては、Ａ）複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶するデータ群記憶領域、Ｂ）前記複数のデータ群の各属性項目に属するデータの存在位置情報を記憶する検索補助データを各属性毎にまとめて記憶する補助データ記憶領域を有する。したがって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶しても、前記検索補助データを用いて、高速検索することができる。

【００１６】７）本発明にかかるコンピュータを用いてデータを検索するデータ検索方法においては、1)複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数、第１の記憶領域に記憶しておき、2)前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを第２の記憶領域に記憶しておき、3)検索する属性項目を含む検索命令が与えられると、前記検索補助データを参照にして、前記第１の記憶領域に記憶されたデータ群を検索する。このように、各属性項目が構造的に定められているデータ群に存在する所望のデータを、前記検索補助データを用いて検索することにより、構造的に構成されたデータ群から高速にデータ検索が可能となる。

【００１７】８）本発明にかかるコンピュータを用いたデータ管理方法においては、複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数第１の記憶領域に記憶しておき、前記複数のデータ群の各属性項目に属するデータの存在位置情報を各属性毎にまとめて記憶する検索補助データを第２の記憶領域に記憶しておき、複数の属性項目についてデータが記憶されたデータ群が新たに与えられると、前記第１の記憶領域に追加記憶するとともに、前記新たに与えられたデータ群の属性項目のうち、前記検索補助データの属性項目と一致する属性項目については、当該属性項目のデータの存在位置情報を追加し、前記検索補助データの属性項目に存在しない属性項目については、あたらな属性項目として追加するとともに、当該属性項目のデータの存在位置情報を前記第２の記憶領域に追加する。したがって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶することができる。また、追加のデータ群を当該構造を崩すことなくすべて記憶することができる。

【００１８】「属性可付加型の記述型データ」とは、実施形態では，ＸＭＬデータを採用したが、これに限定されず、属性が付加できるデータであればどのようなもの

であってもよい。また、属性付加の手法についてはＸＭＬデータのようにタグで指定する方式以外であってもよい。

【００１９】

【発明の実施の形態】１．機能ブロック図の説明

本発明の一実施形態を図面に基づいて説明する。図１に示すデータベース管理システム１は、データ群記憶手段５、補助データ記憶手段７、検索手段３、および追加データ群処理手段９を備えている。

【００２０】データ群記憶手段５は、複数の属性項目についてデータが記憶されたデータ群であって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなく複数記憶する。補助データ記憶手段７は、前記複数のデータ群の各属性項目に属するデータの存在位置情報を属性毎にまとめて記憶する検索補助データを記憶する。検索手段３は、検索する属性項目を含む検索命令が与えられると、補助データ記憶手段７に記憶された検索補助データを参照にして、データ記憶手段５に記憶されたデータを検索する。

【００２１】したがって、構造的に構成されたデータ群から高速にデータ検索が可能となる。また、検索補助データは属性毎にまとめて前記存在位置情報を記憶しているので、より高速検索が可能となる。

【００２２】追加データ群処理手段９は、複数の属性項目についてデータが記憶されたデータ群が新たに与えられると、データ記憶手段５に追加記憶するとともに、前記新たに与えられたデータ群の属性項目のうち、前記検索補助データの属性項目と一致する属性項目については、当該属性項目のデータの存在位置情報を追加し、前記検索補助データの属性項目に存在しない属性項目については、あたらな属性項目として追加するとともに、当該属性項目のデータの存在位置情報を前記補助データ記憶手段７に追加記憶する。したがって、各属性項目が構造的に定められているデータ群を、当該構造を崩すことなくそのまま、複数記憶することができる。また、追加のデータ群を当該構造を崩すことなくすべて記憶することができる。

【００２３】２．ハードウェア構成

図２に、図１に示すデータベース管理システム１を、ＣＰＵを用いて実現したハードウェア構成の一例を示す。データベース管理システム１は、ＣＰＵ２３、メモリ２７、ハードディスク２６、ＦＤＤ２５、モニタ３２、キーボード２８，マウス３１およびバスライン２９を備えている。

【００２４】ＣＰＵ２３は、ハードディスク２６に記憶されたプログラムにしたがいバスライン２９を介して、各部を制御する。このプログラムは、ＦＤＤ２５を介して、プログラムが記憶されたフレキシブルディスク２５ａから読み出されてハードディスク２６にインストールされたものである。なお、フレキシブルディスク以外に、ＣＤ－ＲＯＭ、ＩＣカード等のプログラムを実体的に一体化したコンピュータ可読の記録媒体から、ハードディスクにインストールさせるようにしてもよい。さらに、通信回線を用いてダウンロードするようにしてもよい。

【００２５】本実施形態においては、プログラムをフレキシブルディスクからハードディスク２６にインストールさせることにより、フレキシブルディスクに記憶させたプログラムを間接的にコンピュータに実行させるようにしている。しかし、これに限定されることなく、フレキシブルディスクに記憶させたプログラムをＦＤＤ２５から直接的に実行するようにしてもよい。なお、コンピュータによって、実行可能なプログラムとしては、そのままのインストールするだけで直接実行可能なものはもちろん、一旦他の形態等に変換が必要なもの（例えば、データ圧縮されているものを、解凍する等）、さらには、他のモジュール部分と組合して実行可能なものも含む。

【００２６】ハードディスク２６には、制御プログラムおよびオペレーティングシステム（ＯＳ）を記憶する。オペレーティングシステムとしては、例えば、（株）マイクロソフト社製のＷｉｎｄｏｗｓＮＴ（商標）を採用することができる。

【００２７】ハードディスク２６には、さらに、コンテナボックス２６ｃおよび検索用インデックスデータ２６ｉが記憶されている。コンテナボックス２６ｃには、図３に示すような各データに属性が付加されたＸＭＬデータが、その属性の構造関係を崩さないまま複数記憶される。具体的に説明すると、図３Ａに示すＸＭＬデータは、各属性の構造関係は図４に示すようなツリー構造で表される。かかるツリー構造をそのまま記憶するために、図５に示すように、各属性に関する実データがどの領域に記憶されているかを示すポインタが記憶されている。例えば、属性「名前」については、実データ「山田一郎」の先頭アドレスを示すポインタＰ１が記憶されている。また、本実施形態においては、各属性の構造関係はツリー構造で表されているので、属性「所属」にさらに、属性「部」が所属しており、その実データはポインタＰ５に記憶されており、さらに、この場合、ポインタＰ５で指し示される領域には属性「課」が記憶されており、属性「課」の実データはポインタＰ６で指し示される領域に記憶されている。本実施形態においては、１のＸＭＬデータのデータ群をブロックデータと呼ぶ。コンテナボックス２６ｃには、かかるブロックデータが複数記憶されている。

【００２８】検索用インデックスデータ２６ｉについて、図７を用いて説明する。検索用インデックスデータ２６ｉは、コンテナボックス２６ｃに記憶された複数のブロックデータを検索するためのインデックスデータであり、コンテナボックス２６ｃに記憶された複数のブロッ

クデータの各属性のツリー構造を統合した統合ツリー構造で記憶されている。さらに、統合ツリー構造のすべての属性について、当該属性についてデータを有するブロックデータについて、実データのコンテナidと先頭アドレス（インデックス情報）が属性毎にまとめて記憶されている。

【００２９】統合ツリー構造について説明する。統合ツリー構造とは、全ブロックデータのツリー構造について、ＯＲ演算を行ったことにより、全ブロックデータのツリー構造が統合されたツリー構造をいう。例えば、図６に示すように、図６Ａに示すツリー構造と、図６Ｂに示すツリー構造と図６Ｃに示すツリー構造があった場合、図６Ｄが統合ツリー構造となる。

【００３０】かかる統合ツリー構造の各属性には、当該属性にデータを有するブロックデータの実データのブロックidおよび先頭アドレス（インデックス情報）が記憶されている。例えば、統合ツリー構造は、属性「名前」には、ブロックid＝０のブロックデータにおける属性「名前」の実データ「山田一郎」が格納されているブロックidとコンテナボックスにおけるアドレス０００１、ブロックid＝１のブロックデータにおける属性「名前」の実データ「田中次郎」が格納されているブロックid＝１とコンテナボックスにおけるアドレス００２１・・・・というように、検索用のデータが記憶されている。かかる検索用アドレスの用い方については後述する。

【００３１】なお、統合ツリー構造であるので、あるブロックデータについては、当該属性についてデータを有しない場合もある。例えば、ブロックid＝２のデータが、属性「所属」に属性「総務部」だけで、属性「課」が存在しない場合には、属性「課」には、ブロックid＝２のデータは存在しないので、そのインデックスは記憶されない。

【００３２】このようにコンテナボックス２６cに記憶されたブロックデータは、各属性項目の構造はツリー構造であり、各属性についてデータが存在するブロックデータを検索するためのインデックスデータが属性毎にまとめて記憶されている。

【００３３】メモリ２７には、各種の演算結果等が記憶される。

【００３４】キーボード２８およびマウス３１は各種のデータまたは命令を入力する入力手段である。モニタ３２には検索結果等が表示される。

【００３５】３．フローチャート

（３．１）データ追加記憶処理について

ハードディスク２６へのデータ記憶処理について図８を用いて説明する。データ追加は、コンテナボックス２６cに新たなブロックデータが追加されるとともに、検索用インデックス２６iにインデックスデータが追加される。ここでは、図４に示す統合ツリー構造が既に記憶されている場合に、図９Ａに示すＸＭＬデータが与えられたものとする。ＣＰＵ２３は、与えられたＸＭＬデータをその属性の構造を崩さないままコンテナボックス２６cに、記憶する（ステップＳ１）。これにより、ブロックデータが１つ追加される。

【００３６】ＣＰＵ２３は、かかるＸＭＬデータから図９Ｂに示すツリー構造を抽出する（ステップＳ３）。これはＸＭＬデータのタグの関係構造を抽出することにより実現できる。例えば、図９Ａからタグ「社員」の下位には、タグ「名前」「所属」、「役職」、「電話番号」が位置しており、さらに、タグ「役職」の下位にはタグ「部」が位置しており、さらにタグ「部」の下位にはタグ「課」が位置していることがわかる。これにより、図９Ｂに示すように、各属性のツリー構造が抽出できる。

【００３７】ＣＰＵ２３は、処理番号iを初期化し（ステップＳ５）、i番目の属性を注目属性とする（ステップＳ７）。ＣＰＵ２３は注目属性について、既に統合ツリー構造に存在するか否か判断する（ステップＳ９）。この場合、属性「名前」は既に、図７の統合ツリー構造に存在するので、ステップＳ１３に進み、当該属性に属するデータ「鈴木太郎」の前記コンテナボックスにおける先頭アドレスを、前記統合ツリー構造の属性「名前」に当該ブロックidとともに、記憶する。

【００３８】ＣＰＵ２３は検討していない注目属性が存在するか否か判断し（ステップＳ１５）、存在している場合には、処理番号iをインクリメントし（ステップＳ１７）、ステップＳ７以下の処理を行う。この場合、残りの属性「所属」、「部」、「課」、「役職」、「電話番号」については、すべて既に存在するので、同様にして新しいアドレスデータのみ記憶し（ステップＳ１３）、すべての項目について検討すると処理を終了する。

【００３９】なお、属性が存在するか否かは、その上位の属性が異なる場合には、既に存在しないと判断すればよい。すなわち、完全一致の場合に既に存在すると判断すればよい。例えば、この場合、属性「課」は、属性「部」の下部に位置するが、所属の下に属性「課」が存在する場合には、異なる属性であると判断すればよい。

【００４０】このようにして、新しいＸＭＬデータは、その属性の構造関係を崩すことなく、記憶され、さらに、検索用インデックス２６iに追加記憶される。

【００４１】つぎに、既に存在する属性ではない属性が与えられた場合について、具体例を用いて説明する。図７に示す統合ツリー構造が既に記憶されている場合に、図１０に示すＸＭＬデータが与えられると、ＣＰＵ２３は、与えられたＸＭＬデータをその属性の構造を崩さないままコンテナボックス２６cに、記憶する（図８ステップＳ１）。かかる処理は、存在するか否かに関わりなく同じである。つぎに、ＣＰＵ２３は、かかるＸＭＬデータから図１１に示すツリー構造を抽出する（ステップＳ３）。ＣＰＵ２３は、処理番号iを初期化し（ステッ

プＳ５）、i番目の属性を注目属性とする（ステップＳ７）。ＣＰＵ２３は注目属性について、既に統合ツリー構造に存在するか否か判断する（ステップＳ９）。この場合、属性「名前」は既に、図４の統合ツリー構造に存在するので、ステップＳ１３に進み、当該属性に属するデータ「佐藤五郎」の前記コンテナボックスにおける先頭アドレスを、前記統合ツリー構造の属性「名前」に当該ブロックidとともに、記憶する。

【００４２】ＣＰＵ２３は、属性「所属」、「部」、「課」について、同様の処理を行う。つぎに属性「係」については、図７に示す統合ツリー構造には存在しないので、図８ステップＳ９にて、対象の属性が存在しないとして、新しい属性として統合ツリー構造に追加する（ステップＳ１１）。

【００４３】属性「電話番号（携帯）」についても同様にして、統合ツリー構造に追加される。これにより、統合ツリー構造は図１２に示すように、属性「係」、「電話番号（携帯）」が追加される。

【００４４】（３．２）データ検索処理について

データ検索処理について図１３のフローチャートを用いて説明する。検索命令が与えられると、ＣＰＵ２３は検索用インデックス２６iから統合ツリー構造の構造を取得する（ステップＳ２１）。ここでは、検索命令として、営業部第１課に所属する社員の名前を抽出する検索命令が与えられたものとして説明する。

【００４５】ＣＰＵ２３は、前記統合ツリー構造を参照にして、検索対象の属性は属性「所属」の下位の属性「部」のさらに下位の属性「課」であると判断して、当該属性「課」に記憶されているインデックスデータ（ブロックidおよび先頭アドレス）を読み出す（ステップＳ２３）。図７に示す例であれば、「ブロックid＝０，アドレス０００８」、「ブロックid＝１，アドレス００２０」「ブロックid＝３，アドレス００３５」・・・が読み出される。ＣＰＵ２３はかかるインデックスデータから、コンテナボックス２６cを検索して、実データを読み出す（ステップＳ２５）。

【００４６】ＣＰＵ２３は、読み出された実データのうち、検索条件に合致するものを選択する（ステップＳ２７）。選択されたデータからブロックidを特定して、コンテナボックスから抽出すればよい。具体的には、検索用インデックスデータの前記属性「名前」に記憶されたインデックスデータのブロックidと、前記特定されたブロックidとが一致するブロックについての、先頭アドレスを取得して、コンテナボックスから抽出するようにすればよい。

【００４７】このように、本実施形態においては、前記各属性項目の構造を記憶しておき、ＣＰＵ２３は、各属性項目の構造を参照にして記憶コンテナ部２６cに記憶されたデータを検索する。したがって、ＸＭＬデータについてデータ構造を崩さないまま記憶した場合でも、高速検索が可能となる。

【００４８】また、統合ツリー構造を記憶しているので、ある属性の上位の属性までが検索条件として与えられなくても、検索が可能となる。

【００４９】なお、検索命令として検索対象の属性とともに、その上位の属性が与えられた場合には、ステップＳ２１の処理は不要である。

【００５０】なお、データ削除する場合には、追加とは逆の処理をすればよい。すなわち、コンテナボックスから削除するとともに、検索用インデックスから該当するデータを削除すればよい。

【００５１】また、既に記憶されているブロックデータについて、新しい属性を追加する場合には、一旦該当するデータを破棄して、新たに、かかる属性を追加したＸＭＬデータを与えればよい。しかし、これ以外にも、既に存在する属性およびその値については無視して、追加すべき属性のみブロックデータおよび検索用インデックスに追加するようにしてもよい。例えば、社員「山田一郎」について、新たな属性である属性「携帯電話」を追加する場合には、図１４に示すＸＭＬデータが与えられると、ＣＰＵ２３は、コンテナボックスを検索して、同じブロックが存在するか否か判断して、存在する場合には、異なる属性を抽出して、当該ブロックデータに追加するとともに、検索用インデックスに属性「電話番号（携帯）」を追加すればよい。

【００５２】このように、本実施形態においては、データベースのデータ形式を従来の検索に適した形であるレコード形式ではなく、ＸＭＬデータの構造をそのまま記憶するとともに、各属性毎にデータの存在位置を示す検索用インデックスを採用している。したがって、データ形式の変更等が自由である柔軟なデータベース管理システムを提供することができる。また、各項目の属性がツリー構造となっているデータ群のデータ管理に適したデータベース管理システムを提供することができる。

【００５３】なお、本実施形態においては、属性を構造的に特定されたデータとしてＸＭＬデータを用いた場合について説明したが、属性を構造的に特定されたデータであればそのデータ形式に任意である。

【００５４】なお、前記各プログラムのうち一部を、オペレーティングシステム（ＯＳ）プログラムが実行するようにしてもよい。すなわち、プログラム単独で行っても、オペレーティングシステムと分担して、実現するようにしてもよい。

【００５５】なお、本実施形態においては、図１に示す機能を実現する為に、ＣＰＵ２３を用いソフトウェアによってこれを実現している。しかし、一部をハードウェアで実現してもよい。また、ハードウェアとソフトウェアの棲み分けについては任意の組み合わせが可能である。

【図面の簡単な説明】

13

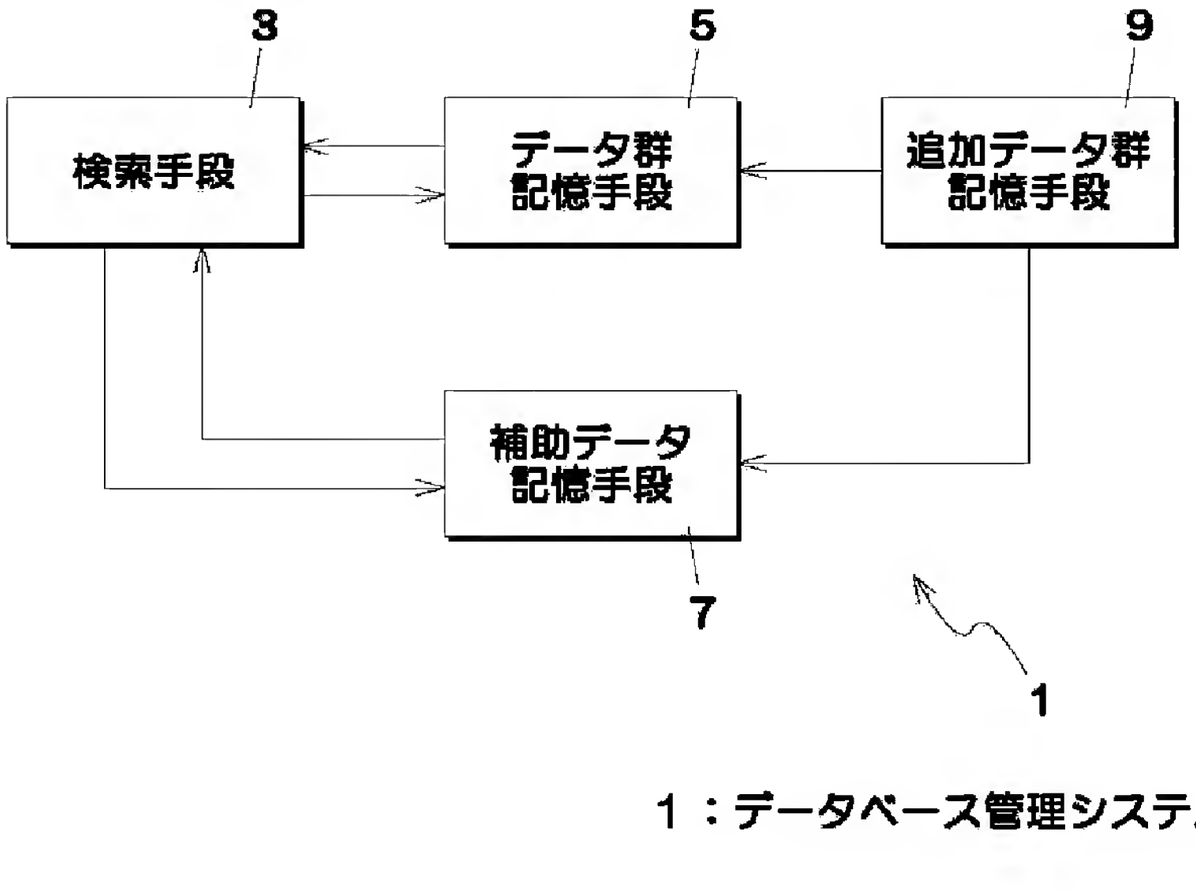
【図１】本発明にかかるデータベース管理システム１の機能ブロック図である。
【図２】データベース管理システム１をＣＰＵを用いて構成したハードウェア構成の一例を示す図である。
【図３】入力されるＸＭＬデータの一例を示す図である。
【図４】図３のＸＭＬデータにおける属性の構造（ツリー構造）示す。
【図５】図３のＸＭＬデータをブロックデータとして記憶した場合の各データの関係を示す。
【図６】統合化ツリー構造を示す図である。
【図７】検索用インデックスデータのデータ構造である。
【図８】データ記憶処理のフローチャートである。
【図９】追加入力されるＸＭＬデータの一例を示す図である。

＊

14

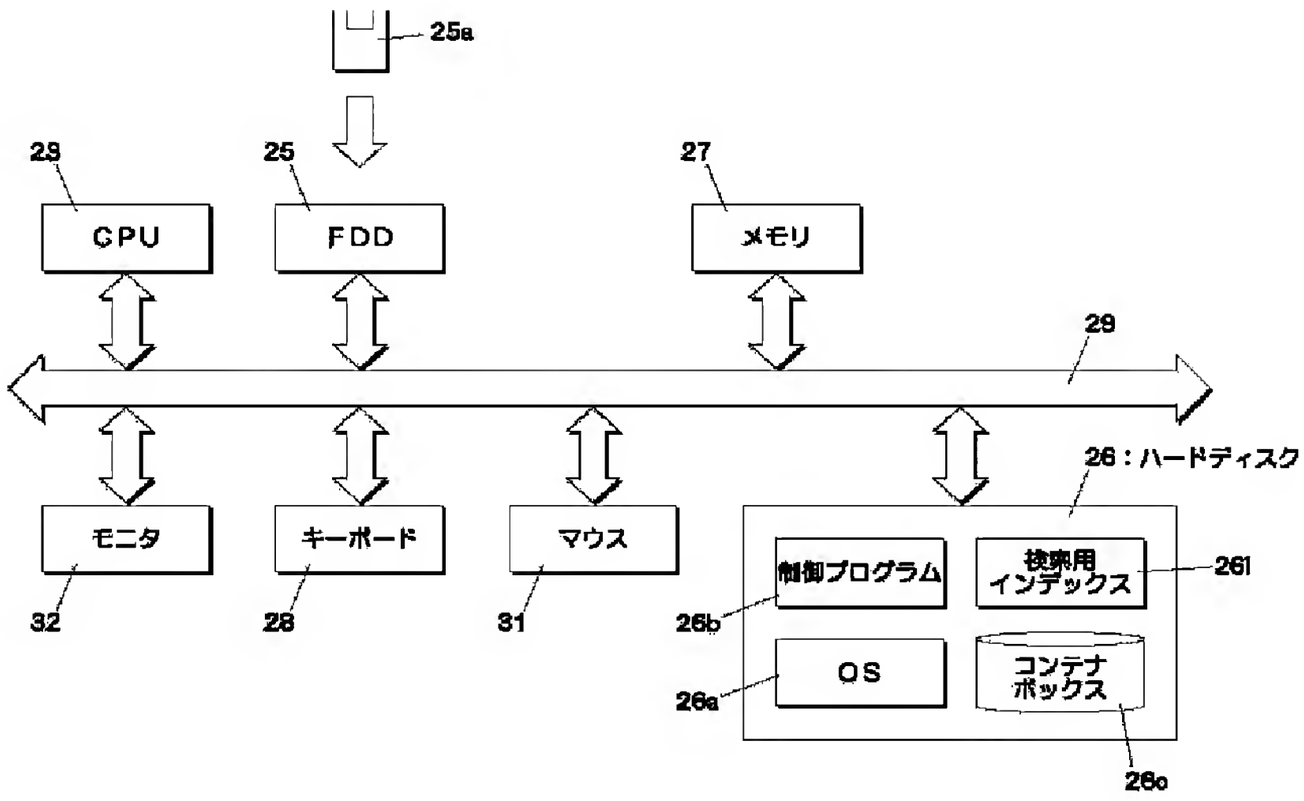
＊【図１０】追加入力されるＸＭＬデータの一例を示す図である。
【図１１】図１０のＸＭＬデータの属性の構造を示す図である。
【図１２】追加後の検索用インデックスデータのデータ構造である。
【図１３】データ検索処理のフローチャートである。
【図１４】追加入力されるＸＭＬデータの一例を示す図である。
【符号の説明】
１・・・データベース管理システム
２３・・・・ＣＰＵ
２６・・・・ハードディスク
２６ｉ・・・・検索用インデックス
２６ｃ・・・・コンテナボックス

【図１】



【図５】

【図２】

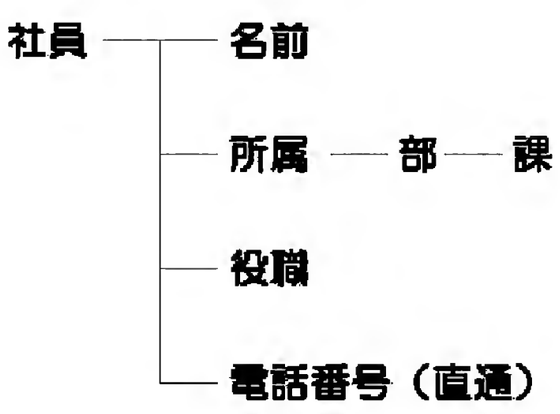


【図３】

A

```
<社員>
    <名前>山田一郎</名前>
    <所属>
        <部>営業部
            <課>一課</課>
        </部>
    </所属>
  <役職>課長</役職>
  <電話番号（直通）>03−1234−4567</電話番号（直通）>
</社員>
```

【図４】



MED00104

B

```
<社員>
    <名前>田中次郎</名前>
    <所属>
        <部>技術部
            <課>二課</課>
        </部>
    </所属>
  <役職>部長</役職>
  <電話番号（直通）>03−1234−4444</電話番号（直通）>
</社員>
```

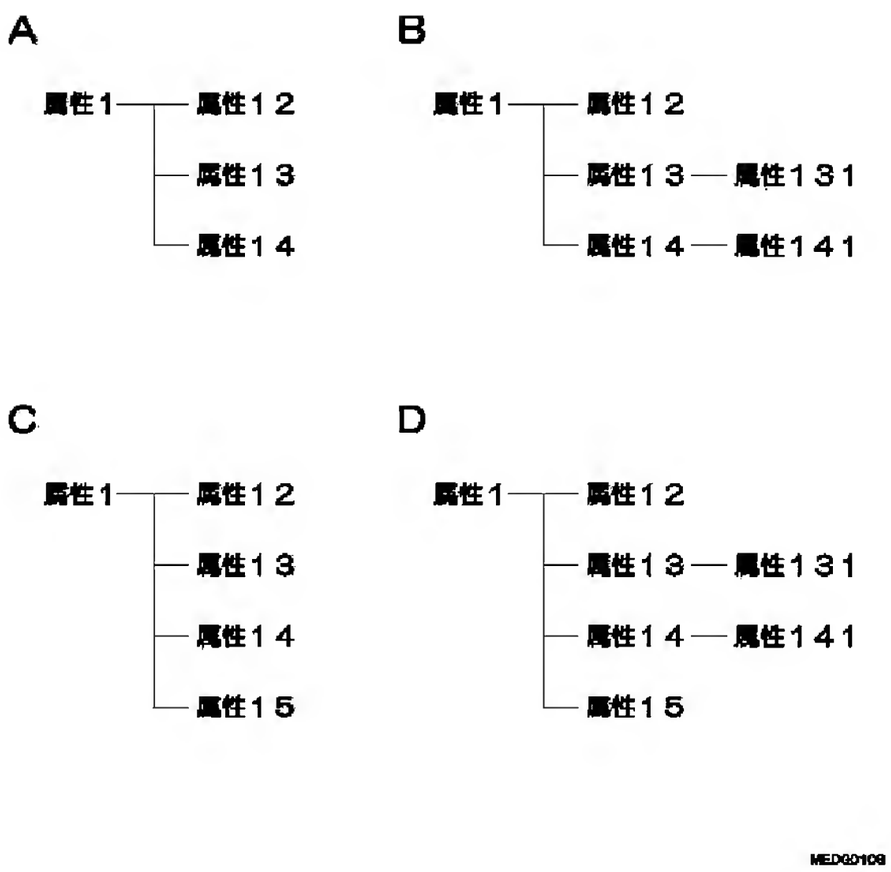
MED00109

【図１０】

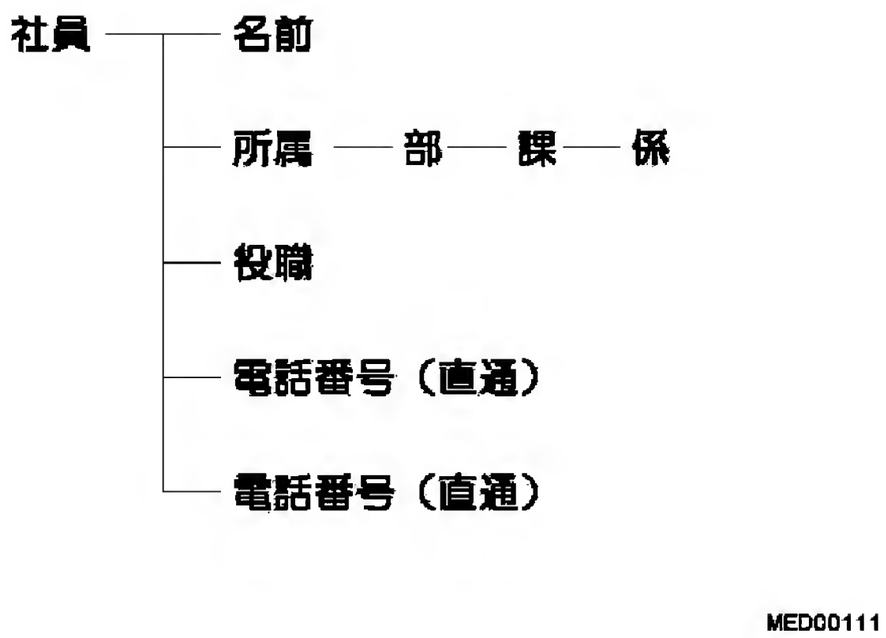
```
<社員>
    <名前>佐藤五郎</名前>
    <所属>
        <部>営業部
            <課>三課
              <係>第一係</係>
            </課>
        </部>
    </所属>
  <役職>主任</役職>
  <電話番号（直通）>03−1234−XXXX</電話番号（直通）>
  <電話番号（携帯）>090−0321−4567</電話番号（携帯）>
</社員>
```

MED00110

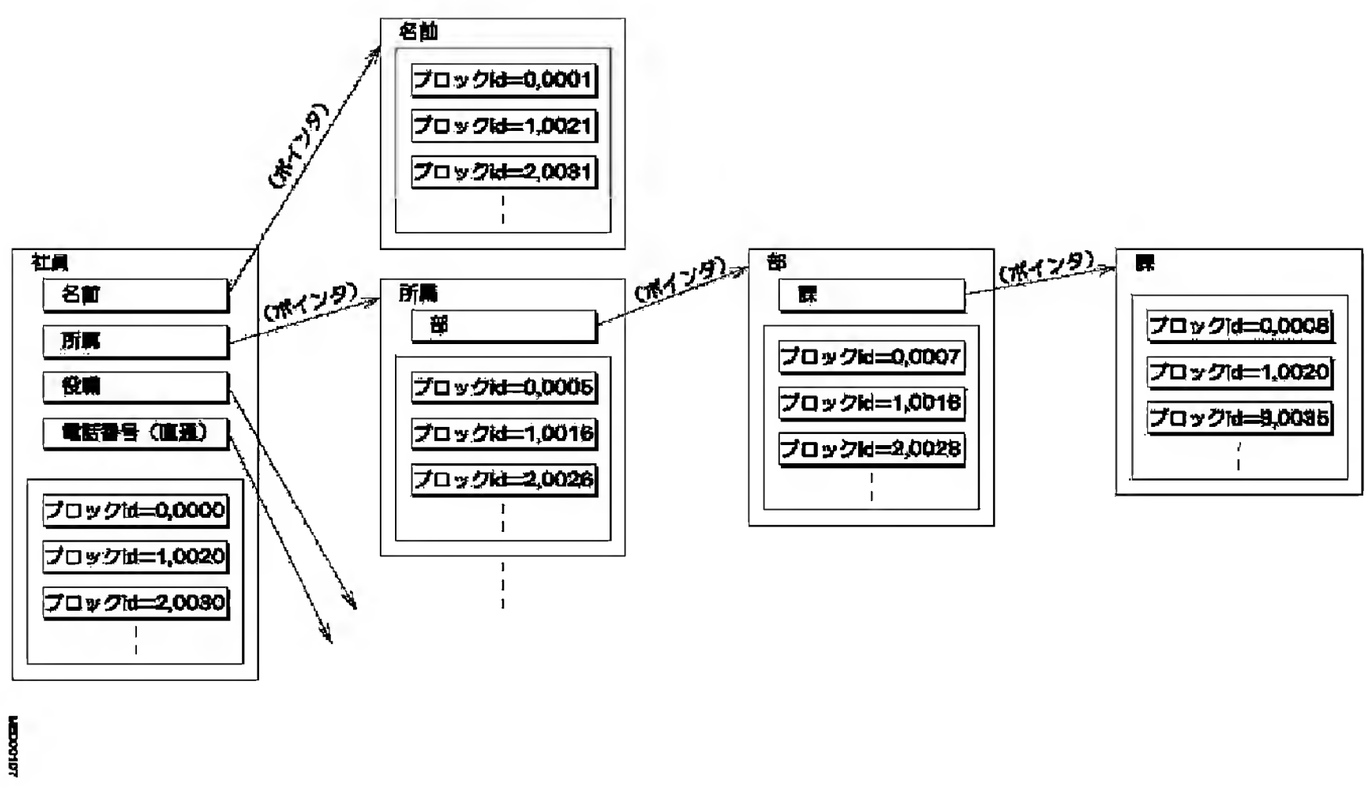
【図６】


A
属性１
属性１２
属性１３
属性１４
B
属性１
属性１２
属性１３
属性１３１
属性１４
属性１４１
C
属性１
属性１２
属性１３
属性１４
属性１５
D
属性１
属性１２
属性１３
属性１３１
属性１４
属性１４１
属性１５
MED0108

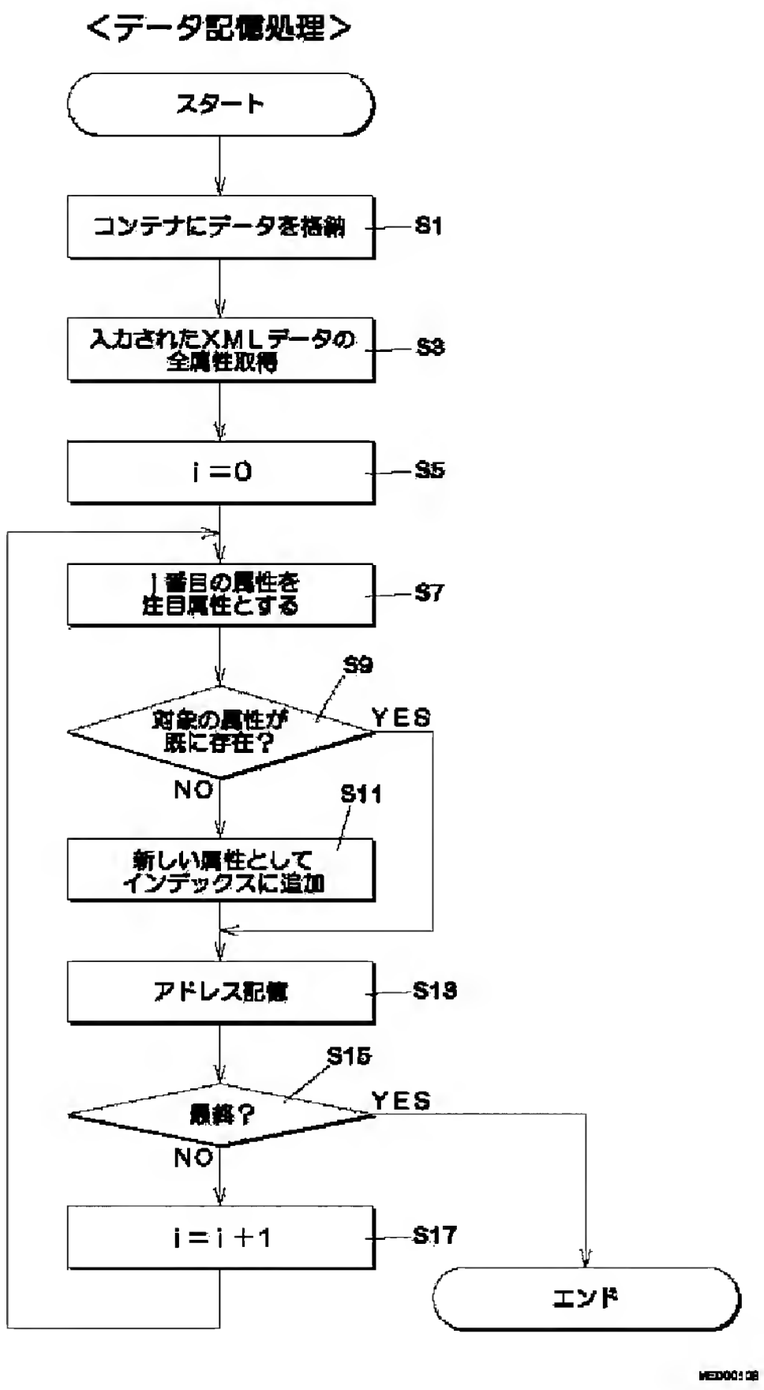
【図１１】


社員
名前
所属
部
課
係
役職
電話番号（直通）
電話番号（直通）
MED0111

【図７】


名前
ブロックId=0,0001
ブロックId=1,0021
ブロックId=2,0031
(ポインタ)
社員
名前
所属
役職
電話番号（直通）
ブロックId=0,0000
ブロックId=1,0020
ブロックId=2,0030
(ポインタ)
所属
部
ブロックId=0,0005
ブロックId=1,0016
ブロックId=2,0026
(ポインタ)
部
課
ブロックId=0,0007
ブロックId=1,0018
ブロックId=2,0028
(ポインタ)
課
ブロックId=0,0008
ブロックId=1,0020
ブロックId=2,0035
MED0107

【図８】

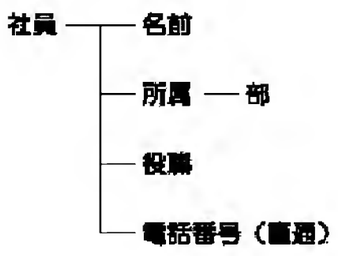


【図９】

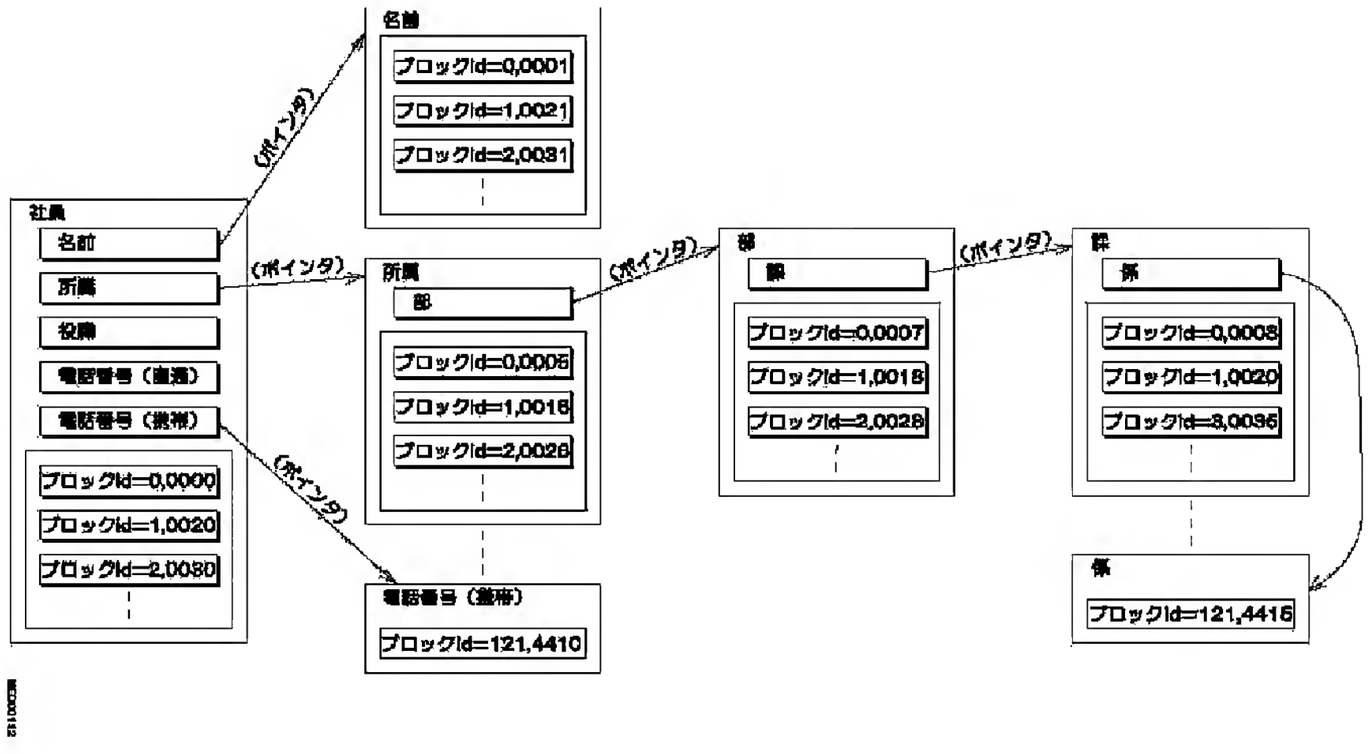
A

```
<社員>
    <名前>鈴木太郎</名前>
    <所属>
        <部>営業部
            <課>三課</課>
        </部>
    </所属>
    <役職>主任</役職>
    <電話番号（直通）>03−1234−XXXX</電話番号（直通）>
</社員>
```

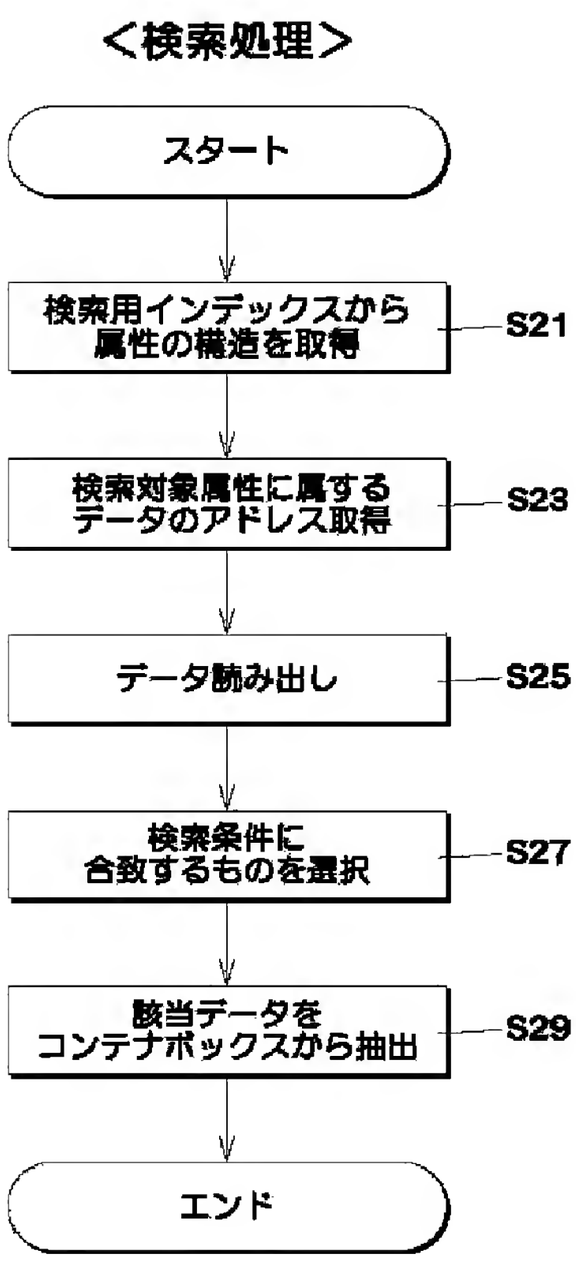
B



【図１２】



【図１３】



MED00113

【図１４】

```
<社員>
    <名前>山田一郎</名前>
    <所属>
        <部>営業部
            <課>一課</課>
        </部>
    </所属>
  <役職>課長</役職>
  <電話番号（直通）>03−1234−4567</電話番号（直通）>
  <電話番号（直通）>090−1234−4444</電話番号（携帯）>
</社員>
```

MED00114

フロントページの続き

Ｆターム(参考)　5B075 NK43 NK46 NR05
5B082 GA07 GC04